APLICACIÓN DE TÉCNICAS DE VISUALIZACIÓN Y ANÁLISIS DE REDES PARA EL ESTUDIO DE LA PRODUCCIÓN COLABORATIVA DE CONOCIMIENTO

DAVID FERNÁNDEZ VILLA

MÁSTER EN INGENIERÍA INFORMÁTICA, FACULTAD DE INFORMÁTICA,
UNIVERSIDAD COMPLUTENSE DE MADRID

Trabajo Fin Máster en Ingeniería Informática

Curso académico 2017/2018

28/08/2018

Director:

Guillermo Jiménez Díaz

APLICACIÓN DE TÉCNICAS DE VISUALIZACIÓN Y ANÁLISIS DE REDES PARA EL ESTUDIO DE LA PRODUCCIÓN COLABORATIVA DE CONOCIMIENTO

DAVID FERNÁNDEZ VILLA

MÁSTER EN INGENIERÍA INFORMÁTICA, FACULTAD DE INFORMÁTICA,
UNIVERSIDAD COMPLUTENSE DE MADRID

Trabajo Fin Máster en Ingeniería Informática

Curso académico 2017/2018 – Convocatoria: Septiembre

Nota: 8

28/08/2018

Director:

Guillermo Jiménez Díaz

Autorización de Difusión

DAVID FERNÁNDEZ VILLA

28/08/2018

El abajo firmante, matriculado en el Máster en Ingeniería Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: APLICACIÓN DE TÉCNICAS DE VISUALIZACIÓN Y ANÁLISIS DE REDES PARA EL ESTUDIO DE LA PRODUCCIÓN COLABORATIVA DE CONOCIMIENTO, realizado durante el curso académico 2017-2018 bajo la dirección de Guillermo Jiménez Díaz en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Resumen en castellano

Entendiendo la existencia de problemáticas que se derivan de la producción de conocimiento colaborativo, es necesario el estudio y análisis en profundidad de este modelo para poder dar solución a estas. En el presente trabajo se realiza el diseño de una herramienta que permita el estudio del comportamiento de los usuarios dentro de una wiki, al ser consideradas estas como comunidades de producción colaborativa. En concreto, se propone desarrollar un prototipo de herramienta que trabaja sobre Wikia, uno de los mayores host de wikis del mundo. Para el desarrollo de la misma se ha realizado una investigación previa sobre el funcionamiento de Wikia y la implicación y utilidad del análisis de redes para el correcto desempeño de la misma. También se explicará de forma concisa las cuestiones más técnicas del desarrollo de la aplicación, siendo complementado con un caso de uso donde se puede apreciar el potencial de la herramienta.

Palabras clave

Wikia, wikis, producción colaborativa, comunidades online, análisis de redes

Resumen en inglés

Understanding the existence of problems that arise from the production of collaborative knowledge, it is necessary to study and in-depth analysis of this model to solve them. In the present work, the design of a tool that allows the study of user behavior within a wiki is done, as these are considered as collaborative production communities. It is proposed to develop a prototype tool that works on Wikia, one of the largest host of wikis in the world. For the development of the same one a previous investigation on the operation of Wikia has been realized and the implication and utility of the analysis of networks for the correct performance of the same one. The more technical questions of the development of the application will also be explained concisely, being complemented with a use case where the potential of the tool can be appreciated.

Keywords

Wikia, wiki, Knowledge P2P production, online communities, network science

Índice de contenidos

Autorización de Difusión iii
Resumen en castellano iv
Palabras clave iv
Resumen en inglés v
Keywords v
Índice de contenidos 1
Agradecimientos 3
Capítulo 1 - Introducción 4
1.1 Objetivos 6
1.2 Metodología de trabajo 6
Chapter 1 – Introduction 8
1.1 Aims 9
1.2 Work methodology 10
Capítulo 2 - Antecedentes e investigación previa 12
2.1 Características de una wiki 13
2.2 Wikia 15
2.2.1 Protección de las páginas 16
2.2.2 Tipos de usuarios 17
2.3 Análisis de redes 20
2.3.1 Propiedades de las redes 21
2.4 Visualización de redes 25
2.5 Conclusiones 27
Capítulo 3 - Diseño de la herramienta 29
3.1 Prototipo de diseño 29
3.2 Extracción de datos 31
3.3 Análisis de los datos 32
3.4 Representación del grafo 35
Capítulo 4 - Desarrollo 37
4.1 Procesamiento de datos y generación del grafo 37

4.2 Visualización del grafo 42
4.2.1 Librerías de visualización 42
4.2.2 Dimensiones visuales 45
4.3 Aplicación web 49
4.3.1 Cliente 50
4.3.2 Servidor 52
Capítulo 5 - Caso de estudio 54
5.1 Limitaciones 58
Capítulo 6 - Conclusiones y trabajo futuro 59
6.1 Trabajo futuro 60
Chapter 6 - Conclusions and future work 61
6.1 Future work 62
Bibliografía 63

Agradecimientos

Este Trabajo de Fin de Máster se ha desarrollado dentro del Grupo de Aplicaciones para la Inteligencia Artificial (Grupo de Investigación UCM reconocido 910494), dentro del proyecto TIN2014-55006-R, financiado por el Ministerio de Economía y competitividad.

Capítulo 1 - Introducción

La aparición de Internet, tal y como lo conocemos actualmente, supuso un gran cambio en la sociedad en lo que va de siglo, permitiendo, entre otras cosas, aumentar la capacidad de coordinación, comunicación y cooperación entre las personas. Son estos los puntos que llevaron a la explosión de la conocida como producción colaborativa o producción entre pares, es decir, a la creación de todo un sistema socioeconómico de generación de bienes materiales y de conocimiento puestos a disposición del común de una manera altruista y voluntaria [1].

Típicamente las redes de trabajo colaborativo se suelen organizar de forma horizontal, generando una gran interdependencia entre los usuarios y siendo cada individuo responsable principal de sus tareas. Esta distribución provee un cambio radical al modelo de producción general basado en la competitividad, ya que en este tipo de redes el objetivo principal es el de compartir el conocimiento en un proceso de aprendizaje mutuo siendo, a priori, más justa y equitativa. Además, este modelo de trabajo ha sido foco de múltiples análisis y estudios a nivel sociológico, tanto a niveles fuera del mundo tecnológico (comunidades educativas, vecinales, huertos colaborativos, etc.) como dentro de Internet, siendo estos últimos de especial interés por varias razones.

Uno de los motivos de interés en el análisis de estas redes de trabajo colaborativo es la abundancia de sitios web de generación colaborativa de conocimiento, como puede ser el caso de las *wikis*, donde los usuarios pueden crear fácilmente contenido y enlazarlos entre sí, siendo el caso más conocido la *Wikipedia*. Otro motivo de interés es la facilidad de cuantificar y recabar datos sobre la participación de los usuarios en la propia red, dado que la propia tecnología nos permite registrar casi cualquier actividad que se realice en la misma, facilitando su posterior análisis.

No obstante, pese a la distribución altruista del conocimiento que propone el modelo sobre la teoría, algunos estudios apuntan a una desigualdad en la producción de contenido [2]. Ya en 2006 Jakob Nielsen propuso una ley bautizada como *90-9-1* [3] en la que se define que el 90% de los usuarios de un sitio web de producción colaborativa únicamente se dedica a observar y leer

contenido sin realizar ningún tipo de contribución, siendo por tanto la inmensa mayoría consumidores. Después estaría el 9% de los usuarios que generarían contenido, pero con una baja frecuencia, siendo, en el mejor de los casos, usuarios que mantuvieran un equilibrio producción-consumo. Finalmente, el 1% restante serían los usuarios que generarían la gran parte del contenido de la red de forma frecuente, recayendo sobre ellos el peso y la responsabilidad del funcionamiento de la red.

Sin entrar a valorar las posibles causas y soluciones a este problema, pues no es la labor que nos compete, basta ver que es un serio problema que cuestiona y pone en jaque el funcionamiento de todo un modelo de producción y, por ende, a una buena parte de Internet y su modelo de compartición de conocimiento.

Por tanto, con el fin de facilitar el estudio y análisis de problemáticas como esta, el presente Trabajo Fin de Máster tiene como objetivo principal crear una herramienta de visualización y análisis de redes que será prototipada para funcionar con Wikia, uno de los sitios web más grandes de producción colaborativa de conocimiento.

Se puede pensar que Wikipedia al ser, por tamaño y popularidad, el mayor sitio web de producción de conocimiento colaborativo es la mejor opción sobre la que basar el funcionamiento de la herramienta. No obstante, existen ciertas objeciones para su elección, siendo uno de ellos la existencia de numerosos estudios y herramientas centradas en Wikipedia y el hecho de ser una red excesivamente grande y difícil de dividir para según qué análisis se quieran realizar. Es por esto por lo que viene motivada la elección de Wikia, siendo uno de sus puntos a favor la existencia de wikis independientes de diverso tamaño que permite el estudio del comportamiento en función de sus dimensiones. Además, otra de las motivaciones en la elección es la existencia de proyectos paralelos sobre el funcionamiento y comportamientos en Wikia dentro de la propia Facultad de Informática realizados por compañeros que pueden utilizar la herramienta para futuras investigaciones y desarrollos.

1.1 Objetivos

El principal fin de este proyecto es desarrollar una herramienta de visualización y análisis de redes para estudiar sistemas de producción colaborativa, de la que se desarrollará un prototipo con un funcionamiento adaptado para usar con Wikia. Como objetivo se pretende que la herramienta proporcione distintas métricas y permita de una forma sencilla comprender como es el comportamiento y la interacción de los usuarios dentro de la misma.

Además del objetivo principal, existen una serie de objetivos secundarios derivados del desarrollo del propio TFM que son los siguientes:

1. Investigar sobre Wikia, su funcionamiento, su API, los tipos de usuarios existentes y datos que puedan resultar de especial relevancia en el desarrollo del proyecto.

2. Ampliar los conocimientos básicos sobre grafos y Análisis de Redes Sociales, teniendo en cuenta las técnicas y métricas que pueden resultar interesantes para este trabajo.

3. Investigar y aprender el uso de herramientas y librerías que permitan el tratamiento de datos y la generación, análisis y visualización de grafos a partir de los mismos.

4. Investigar el uso de tecnologías web que resulten de especial utilidad y posterior diseño del sistema que facilite el uso de la propia herramienta.

1.2 Metodología de trabajo

La metodología seguida para este trabajo fue de investigación para la fase previa y un desarrollo incremental para el resto del proyecto. En primer lugar, se hizo un estudio previo de la situación actual de los campos, es decir, el funcionamiento de las wikis en general y de Wikia en particular, así como el análisis de redes y las posibilidades que brinda.

Una vez realizada la investigación, se continuó con el diseño de la aplicación y como adaptar la teoría obtenida a las necesidades de desarrollo de la herramienta, es decir, como aplicar lo aprendido sobre el funcionamiento de Wikia y las técnicas ofrecidas por el análisis de redes en el desarrollo. Posteriormente, se comenzó la creación de la aplicación, separada en tres fases secuenciales donde la idea es desarrollar en primera instancia el núcleo de la aplicación con unas

funcionalidades limitadas, pero suficientes para obtener el grafo asociado a la wiki y distintas medidas sobre este. Después, en una segunda fase se incrementó la funcionalidad permitiendo la visualización del grafo y la interacción por parte del usuario. En la fase final se unificaron los módulos desarrollados en las anteriores fases bajo una aplicación web que facilite su acceso y uso.

La estructura del presente documento mantiene un formato similar a los procesos realizados para la consecución del proyecto. En un primer lugar, en el siguiente capítulo, se expondrán los resultados de una investigación inicial sobre la situación del campo de trabajo y de posibles herramientas que puedan servir de ayuda para abordar el desarrollo del proyecto. Posteriormente, en el capítulo 3, se entrará en detalle sobre el diseño y funcionalidad de la herramienta a desarrollar. Este diseño tomará forma en el capítulo 4 en el que se irán desgranando las cuestiones más técnicas sobre el desarrollo realizado. Finalmente se concluirá con un par de capítulos en los que se muestra un caso de uso de la herramienta (capítulo 5) y se cierre con las conclusiones obtenidas y el trabajo futuro a desarrollar (capítulo 6).

Chapter 1 – Introduction

The emergence of the Internet, as we know it today, was a great change in society so far this century, allowing to increase the ability of coordination, communication and cooperation between people. These are the main points that led to the explosion of what is known as collaborative production or peer-to-peer production, that is, the creation of a whole socio-economic system for the generation of material goods and knowledge made available to the common in an altruistic and voluntary way [1].

Typically, collaborative work networks are usually horizontally organized, generating great interdependence among users and everyone being primarily responsible for their tasks. This distribution provides a radical change to the general production model based on competitiveness, since in this type of networks the main objective is to share knowledge in a process of mutual learning, being, a priori, fairer and more equitable. In addition, this work model has been the focus of multiple analyzes and studies from a sociological perspective, both at levels outside the technological world (educational communities, neighborhoods, collaborative gardens, etc.) and within the Internet, the latter being of special interest for several reasons.

One of these reasons is the huge amount of collaborative knowledge generation websites, such as wikis, where users can easily create content and link it to each other, being Wikipedia the most well-known example. Another reason of interest is the ease of quantifying and collecting data on the participation of users in the network itself, given the fact that technology allows us to record almost any activity that takes place in it, helping further analysis.

However, despite the altruistic distribution of knowledge proposed by the mentioned model, some studies point to inequality in the production of content [2]. Back in 2006 Jakob Nielsen proposed a law named 90-9-1 [3] where it is defined that 90% of the users of a collaborative production website only use it to observe and read content without making any kind of contribution, being therefore the clear majority consumers. Then there would be 9% of the users that would generate content, but with a low frequency, being, in the best of cases, users that

maintained a production-consumption balance. Finally, the remaining 1% would be the users that would generate a large part of the network's content on a frequent basis, with the weight and responsibility of the operation of the network falling on them.

Putting aside the possible causes and solutions to this problem, because it is not the issue that concerns us, it is enough to say that it is a serious problem that questions and puts in check the functioning of a whole production model and, therefore, a good part of the Internet and its model of knowledge sharing.

Therefore, in order to facilitate the study and analysis of problems such as this, this Master's Thesis aims to create a tool for visualization and analysis of networks within Wikia, one of the largest collaborative production websites of knowledge.

One can think that, being Wikipedia the largest collaborative knowledge production website by size and popularity, it would be the best option on which base the function of the tool. However, there are some objections to their choice, one of them being the existence of numerous studies and tools centered on Wikipedia and the fact that it is an excessively large network and difficult to divide according to the type of analysis that might be done. This is the reason why Wikia has been choseen, being the existence of independent wikis of different sizes that allow the study of behavior based on its dimensions one of its points in favor. In addition, another of the motivations in the choice is the existence of parallel projects about the functioning and behaviors in Wikia within the Faculty of Computing made by colleagues who can use the tool for future research and development.

1.1 Aims

The main purpose of this project is to develop a network visualization and analysis tool in order to study collaborative production systems, for which a prototype will be developed for use with Wikia. The objective is that the tool provides different metrics and allows a simple way to understand how the behavior and interaction of users within it is.

In addition to the main objective, there are a series of secondary objectives derived from the development of the TFM itself, which are the following:

1. To investigate Wikia, its operation, its API, the types of existing users and data that may be of special relevance in the development of the project.
2. To expand basic knowledge about graphs and Social Network Analysis, considering the techniques and metrics that may be interesting for this work.
3. To investigate and learn the use of tools and libraries that allow the processing of data and the generation, analysis and visualization of graphs from them.
4. To investigate the use of web technologies that are especially useful and subsequent design of the system that facilitates the use of the tool itself.

1.2 Work methodology

The methodology followed for this work was research for the previous phase and an incremental development for the rest of the project. In the first place, a previous study of the current situation of the fields, that is, the functioning of wikis in general and of Wikia in particular, as well as the analysis of networks and the possibilities that it offers, was made.

Once the research was done, we started with the design of the application focusing on how to adapt the theory obtained to the tool's development needs, that is, how to apply what has been learned about the functioning of Wikia and the techniques offered by network analysis. Subsequently, the creation of the application began, separated into three sequential phases where the idea was to develop the core of the application in the first instance with limited functionalities, but sufficient to obtain the graph associated with the wiki and various measures on it. Then, in a second phase, the functionality was increased, allowing the visualization of the graph and the interaction by the user. In the final phase, the modules developed in the previous phases were unified under a web application that facilitates their access and use.

The structure of this document maintains a format like the processes carried out to achieve the project. In the first place, in the next chapter, the results of an initial investigation about the field of work and of possible tools that can help to approach the development of the project will be exposed. Later, in chapter 3, we will go into detail about the design and functionality of the tool

to be developed. This design will take shape in chapter 4 in which the most technical questions about the development carried out will be unraveled. Finally, it will conclude with a couple of chapters that show a case of use of the tool (chapter 5) and close with the conclusions obtained and the future work to develop (chapter 6).

Capítulo 2 - Antecedentes e investigación previa

En este capítulo se describirán los resultados de la investigación inicial sobre cuestiones relacionadas con el desarrollo del proyecto, desde qué es una wiki y cómo funciona a la importancia del análisis de redes para el estudio de estas. Además, veremos que este campo nos aporta medios suficientes para sacar métricas y visualizaciones que serán de gran ayuda para comprender el comportamiento dentro de las wikis.

Para comprender qué es una wiki, primero debemos entender que Internet es una fuente enorme de producción de contenido. Sin embargo, todo este conocimiento no puede ni debe ser considerado como parte de la producción colaborativa. En primer lugar, porque una buena parte del contenido generado en Internet viene motivado por el lucro, contradiciendo el espíritu altruista de este movimiento. En otros casos, la generación de conocimiento se realiza de una manera totalmente descoordinada y carente de una cohesión como comunidad como, por ejemplo, en los blogs donde la gente comparte conocimiento de manera independiente y sin tener ningún tipo de coordinación con la comunidad.

Sin embargo, como ya se comentó en el anterior capítulo, existe un tipo de sitio web que combina todos los requisitos de la producción colaborativa que son las *wikis*. Este concepto fue acuñado por Howard Cunningham en 1995, quién también creo el primer sitio *wiki* la WikiWikiWeb[1]. Las características principales de una wiki fueron definidas por el propio Cunningham, junto con Bo Leuf [4], en 2001, siendo estas:

- Cualquier visitante de una wiki puede ser también editor y creador de contenido sin necesidad de grandes conocimientos técnicos ni herramientas extra más allá del propio navegador.
- Debe ser posible enlazar contenido entre distintas páginas de la wiki de una forma sencilla e intuitiva para el usuario.

[1] http://wiki.c2.com/?WikiHistory

Para conseguir crear un sitio web de estas características es necesario disponer de un software especial que se ejecute sobre el propio servidor web. Hoy en día existe una amplia oferta de este tipo de aplicaciones[2], escritos en distintos lenguajes y con diferentes características para adaptarse a diversas necesidades. De entre todos, hay dos que resaltan especialmente: WikiWikiWeb, software que sustenta la wiki homónima, por ser el primero de este tipo; y MediaWiki[3] por ser el más popularizado ya que da soporte a la Wikipedia y Wikia, entre otras. Dada la relevancia de este último, será el que se emplee para ilustrar cómo es el funcionamiento de una wiki.

2.1 Características de una wiki

Una de las características de toda wiki es poder ser editada y crear contenido por usuarios sin grandes conocimientos técnico. Para lograrlo es necesario utilizar un lenguaje que se aleje de los clásicos lenguajes de programación. Es por esto por lo que se crearon lenguajes de marcado que fuesen intuitivos y fáciles de aprender o, en el caso de algunos software, se proporcionan editores que permitan diseñar la página tal y como quedaría finalmente. Normalmente, el contenido generado va asociado a una hoja de estilos común a todas las páginas dando una cierta uniformidad a toda la web. Un ejemplo ilustrativo de cómo de simple es este lenguaje de marcado, lo vemos en la creación de encabezados en el software MediaWiki. El siguiente texto[4]:

```
= Heading 1 =
== Heading 2 ==
=== Heading 3 ===
==== Heading 4 ====
===== Heading 5 =====
====== Heading 6 ======
```

Daría como resultado los encabezados que se observan en la Figura 2.1, independientemente en que página se cree, permitiendo de una forma sencilla y con unas reglas básicas mantener un mismo estilo en todas las páginas. Además, este lenguaje de marcado permite

[2] https://en.wikipedia.org/wiki/List_of_wiki_software

[3] https://www.mediawiki.org/wiki/MediaWiki

[4] Ejemplo extraído de https://en.wikipedia.org/wiki/Help:Wikitext#Sections

fácilmente cumplir con la otra característica de las wikis, poder referencias otras páginas de manera intuitiva. El siguiente ejemplo ilustra cómo se hace:

```
London has [[public transport]].
```

Este texto, en el software MediaWiki, daría como resultado un contenido en el que las palabras *public transport* serían el link a la página homónima donde, se supone, se explicará ese concepto con mayor profundidad. Además, MediaWiki ofrece opciones para que el texto del link no tenga que coincidir exactamente con el nombre de la página que apunta, aunque en el código interno si lo haga.

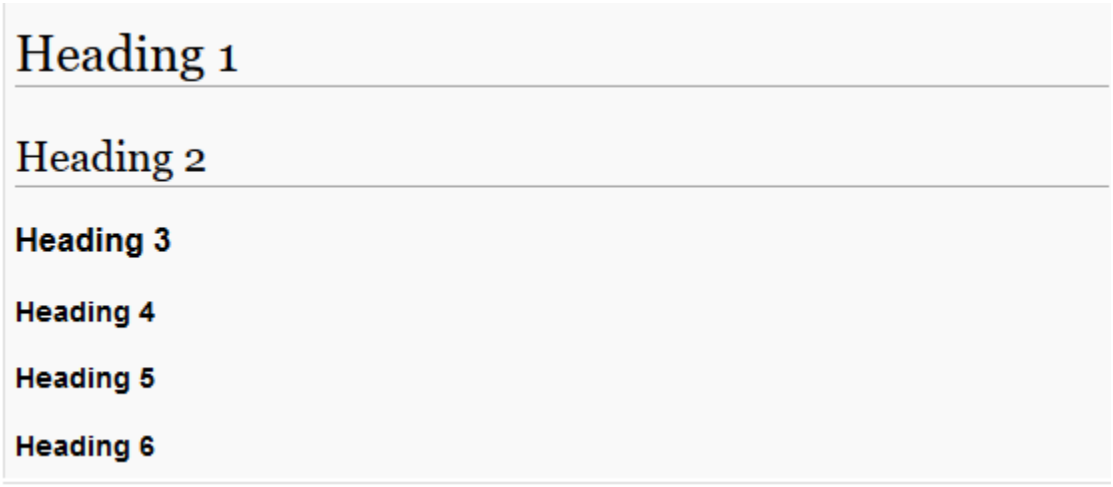

Figura 2.1 Encabezados en MediaWiki

Con esto se obtiene un sitio web con contenido creado y editado por los usuarios que se referencia a sí mismo, facilitando la búsqueda de conceptos asociados a la temática tratada por una página. Aunque en la idea general de la wiki se encuentra la libre creación de contenido y la autocorrección por parte de los usuarios, no siempre es así. En primer lugar, nos encontramos con diferencias entre usuarios sobre la veracidad y exactitud de cierto contenido. Para solventarlo la mayoría de software, y en concreto MediaWiki, proporcionan medios para poder hacer referencias a elementos externos de la wiki que permitan justificar el origen de los datos expuestos en las páginas, siendo esta una buena práctica para evitar las confrontaciones. No obstante, aun con referencias externas, pueden existir debates sobre la organización y el contenido de las páginas.

Como solución se suele proporcionar zonas de debate dentro de las propias páginas para que los usuarios expongan sus opiniones y razones, siendo necesario, en algunos casos, la existencia de mediadores o moderadores.

No obstante, aun con todo lo visto, las páginas pueden ser objeto de vandalismo o cambios que no se amoldan a las normas establecidas o acordadas. Para poder prevenir este tipo de problemas y algún otro como, por ejemplo, fallos en la edición del contenido, se provee de un registro de ediciones. La idea consiste en almacenar un historial en el que quede constancia de cada una de las ediciones que se realizaron sobre una página. De cada edición se almacenarán datos de interés como la fecha, el autor y el contenido del cambio realizado. Este registro permitiría revertir fácilmente ciertas modificaciones o, incluso, expulsar a ciertos usuarios malintencionados.

2.2 Wikia

En lo relativo al presente TFM, el funcionamiento de la herramienta se basa en Wikia, uno de los mayores host de wikis del mundo y el más visitado si nos basamos en el ranking de Alexa[5]. Por este motivo hizo falta una investigación previa sobre la misma que permitiera obtener detalles y puntos de interés, así como el funcionamiento de su API. Cabe destacar que parte de esta investigación se llevó a cabo gracias, en parte, a las herramientas[6] y datos proporcionados por otros investigadores del Departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática

Para empezar, Wikia, renombrada en 2016 como *FANDOM powered by Wikia*[7], es un host de wikis basado en el software MediaWiki, el mismo que Wikipedia, que permite a cualquier usuario crear de forma gratuita y bastante intuitiva una wiki de la temática que se desee. Una de las virtudes de Wikia es la total independencia entre wikis, ya que ni siquiera es obligatorio que la temática de la wiki sea novedosa dentro del universo Wikia, es decir, se pueden repetir temáticas ya existentes. Si bien puede resultar un poco molesto para el usuario final, ya que puede encontrarse con diferentes wikis con distinto contenido sobre un mismo tema, resulta beneficioso

[5] https://www.alexa.com/siteinfo/wikia.com

[6] https://github.com/Grasia/wiki-scripts/

[7] http://comunidad.wikia.com/wiki/Usuario_Blog:Luchofigo85/Wikia_es_ahora_Fandom_powered_by_Wikia

para los usuarios creadores de contenido por la libertad que brinda a la hora de crear wikis o, incluso, de crear variantes de otras ya existentes. Esto supone una diferencia notoria respecto a Wikipedia, donde los límites de cada temática y wiki no están claramente definidos, siendo una comunidad grande y consolidada con subcomunidades integradas.

En lo respectivo al tamaño, actualmente Wikia cuenta con más de 400.000 wikis distintas si nos atenemos a los datos que proporciona su web[8]. No obstante, estos datos suelen ser poco precisos y no permiten saber si cuentan wikis activas o wikis totales almacenadas. Para conseguir obtener un tamaño más preciso se utilizó una de las herramientas desarrolladas por los compañeros que, a partir del *sitemap*[9] de Wikia, obtiene las urls de todas las wikis disponibles, que no tiene por qué ser todas las wikis activas. Una vez obtenido el índice de todas las urls disponibles, mediante otro script se comprueban que las wikis estén activas y, en caso de no estarlo, se eliminan del índice. Con esto se obtuvo que en marzo de 2018 Wikia cuenta con 338000 wikis activas, una cifra que, aunque menor, demuestra que nos encontramos ante una web de grandes dimensiones y gran valor añadido para nuestra herramienta y para el estudio [5].

2.2.1 Protección de las páginas

Para poder comprender mejor el funcionamiento de Wikia y los tipos de usuarios existentes será necesario explicar primero quién y cómo puede editar las páginas. Para ello, se establecen unos niveles de protección de la página que determinará su interacción con los usuarios. Existen tres tipos de protección:

- **Desprotegidas:** Cualquier usuario puede editar y modificar la página, incluso los usuarios no registrados.
- **Semi-protegida:** Protege la página de los usuarios no registrados y los no autoconfirmados, es decir, de los niveles más bajos de usuarios.
- **Totalmente protegida:** La página puede ser editada y movida únicamente por los usuarios administradores y por los moderadores de contenido.

[8] http://www.wikia.com/about

[9] http://www.wikia.com/Sitemap

Además de estos niveles de protección general, se pueden aplicar niveles distintos para según qué acción se quiera realizar. Siendo las acciones posibles editar, mover la página, subir archivos o crear nuevas páginas. Salvo en el caso de páginas especiales, los niveles de protección se eligen a nivel local de la wiki, siendo los administradores quienes decidan el nivel a aplicar.

2.2.2 Tipos de usuarios

En lo relativo a los tipos de usuarios, en nuestro caso, mostraremos especial interés en los usuarios locales de cada wiki, dado que los usuarios globales tienen derechos y funciones enfocados a controlar el funcionamiento general de Wikia como host, así como de velar por el control de spam dentro de las wikis, realizar mejoras técnicas en la plataforma y, en casos excepcionales, servir como mediadores o jueces en disputas entre usuarios.

Entre los usuarios locales nos encontramos con los siguientes grupos y sus privilegios:

Usuarios registrados

Son los usuarios que crearon una cuenta y se registraron en la wiki. Sus privilegios son:

- Personalizar la apariencia y características de la comunidad
- Subir imágenes, video u otros archivos
- Añadir páginas a su ‘watchlist’
- Mantenimiento de un perfil de usuario
- Eliminar anuncios de todas las páginas

Usuarios autoconfirmados

Los usuarios que estén registrados por, al menos, 4 días se considerarán autoconfirmados y obtendrán, además de los privilegios como usuarios registrados, los siguientes:

- No tendrán que introducir un captcha cuando inserten links externos o borren el contenido de páginas.
- Podrán editar páginas semi-protegidas.
- Podrán mover páginas.

Administradores

Son conocidos también como "*admins*" o "*sysops*" y son usuarios confiables que son, normalmente, elegidos por la comunidad para obtener los siguientes privilegios:

- Todos los privilegios de los grupos "Moderador de discusiones" y "Moderador de contenido".
- Bloquear usuarios que tengan un comportamiento inadecuado en la wiki. (Nota: no tienen derecho a desbloquear al usuario)
- Dar y revocar privilegios de "Moderador de chat" y de "Moderador de discusiones"
- Editar el aspecto y formato de la comunidad.
- Editar la lista blanca de páginas de MediaWiki.

Bureaucrats

Están en un nivel superior a los administradores, básicamente, tienen sus mismos privilegios más los siguientes:

- Pueden desbloquear usuarios que hayan sido bloqueados.
- Pueden manipular los privilegios de cada usuario de la wiki, excepto de otros Bureaucrats.
- Pueden dar y revocar permisos de "Rollback", "Moderador de contenido", "Administrador" y de "Bureaucrats" (este último no lo pueden revocar, salvo para sí mismos).
- Pueden revocar estatus de "Bot" cuando exista malfuncionamiento y avisar al Staff para que lo revise.

Además, hay que añadir que el estatus de *bureaucrat* solo puede ser revocado por el Staff o por el propio usuario.

Moderadores de contenido

Son usuarios que tienen adicionalmente los siguientes privilegios para moderar partes de la comunidad:

- Borrar y mover páginas protegidas y archivos.
- Deshacer el borrado de páginas y archivos.
- Rollback

- Resubir archivos eliminados.
- Proteger o desproteger páginas

Moderadores de discusión

Son usuarios que tienen adicionalmente los siguientes privilegios para gestionar conversaciones, a lo largo de toda la comunidad, donde usuarios tengan discusiones. Estos privilegios son:

- Eliminar y restaurar hilos y réplicas de cualquier usuario.
- Cerrar y reabrir hilos
- Gestionar el orden del foro
- Moderar chats
- Eliminar comentarios del blog
- Editar y borrar comentarios en artículos

Moderadores de chat

Son usuarios que tienen el privilegio de expulsar a un usuario de un chat. Este privilegio viene asociado con el deber, como el nombre indica, de moderar los chats de la wiki con tal de facilitar la comunicación y promover el buen comportamiento dentro de la comunidad.

Rollback

Permite al usuario realizar un *rollback* (revertir contenido) a partir del historial y los cambios recientes. Si bien cualquier usuario con derecho a editar una página puede deshacer cambios, los usuarios rollback tienen la facilidad de revertir los cambios con unos pocos clicks, pues con este grupo de privilegios viene asociado el deber de controlar los actos de "vandalismo" que se realicen en la wiki.

Fundador

Este tipo especial de usuario es asignado al creador de la wiki y automáticamente tiene derechos de administrador y *bureaucrat*, siendo, típicamente, el que reparta los roles iniciales dentro de la wiki, así como determine los niveles de protección por defecto de las páginas y el estilo de la wiki.

Bots

Este tipo de usuario especial es el que se les da a los procesos automatizados que se dedican a hacer distintas labores de mantenimiento o de control de la wiki. Son de especial relevancia para el objetivo de nuestra herramienta, pues son usuarios que, al no representar a personas físicas, deberán ser, como poco, tratadas de forma distinta ya que pueden generar ruido en el análisis de los comportamientos que no es deseado.

Usuarios anónimos

Si bien este grupo de usuario no está definido como tal, engloba a los usuarios que acceden y editan contenido sin estar registrados. Esto se puede en los casos en que la wiki no tenga las páginas protegidas. Además, al igual que los bots, estos usuarios suponen un reto para el desarrollo de la herramienta y deberán ser tratados de manera especial, ya que al no estar registrados no cuentan con un ID o nombre de usuario que lo identifique y su comportamiento puede resultar anómalo respecto al resto de usuarios.

2.3 Análisis de redes

Una vez conocido el funcionamiento interno de Wikia es necesario estudiar cómo se puede extraer información de los datos que nos aporta. Para ello cabe recurrir al análisis de las redes o análisis de sistemas complejos, un campo amplio y con un gran impacto, especialmente desde la irrupción de Internet actual, donde es fácil encontrar documentación y estudios sobre cómo la teoría de grafos, con sus métricas y procedimientos, tiene un uso e interpretación dentro del estudio de las redes [6, 7, 8].

Para entender la importancia de este campo, debemos comprender que las redes se encuentran en todas partes, desde Internet a las redes genéticas, pasando por la economía y por una amplia variedad de campos, siendo el análisis de redes, por tanto, un campo multidisciplinar [8]. Entre sus funciones se encuentra el análisis de la estructura y el comportamiento de las redes que permite, entre otros, crear patrones y predicciones de funcionamiento. Otros de los objetivos que persigue el análisis de redes es tratar de extraer información de la red y poder visualizarla de forma que facilite su comprensión. Un ejemplo recurrente de la aplicación del análisis de redes y de sus usos más populares, es la posibilidad de predecir y, por tanto, prevenir, la propagación de

una enfermedad capaz de causar una epidemia mediante el estudio de diversas redes como, por ejemplo, la red de transporte.

Para lograr todo este tipo de estudio sobre las redes y sistemas complejos, el análisis de redes bebe directamente de la teoría de grafos y sus modelos matemáticos. Es por ello por lo que la mayoría de las métricas aplicables a los grafos tienen una interpretación en la red equivalente, siendo de especial relevancia para el propósito de este proyecto. Entre las medidas disponibles, resultan interesantes aquellas que estudian la centralidad de un nodo dentro de la red, es decir, como de importante resulta la posición de este, y aquellas métricas que cuantifican la cohesión de la misma. Por tanto, el proceso seguido para realizar la selección de métricas fue crear un listado con las medidas más populares y de mayor importancia que puedan tener repercusión en el proyecto.

2.3.1 Propiedades de las redes

La mayoría de las métricas que nos ofrecen una cuantificación de los rasgos que buscamos de la red son las dedicadas a medir la centralidad de un nodo. Entendemos por centralidad la capacidad de un nodo para acceder con mayor facilidad al resto de nodos y, por tanto, tener una mayor influencia sobre la comunidad. Si interpretamos esto aplicándolo a nuestro caso significaría que un usuario (siendo los nodos los usuarios) que tenga una mayor centralidad tiene un mayor peso dentro de la comunidad y, por tanto, una mayor importancia. Además, dentro de las métricas de centralidad podemos diferenciar entre aquellas que la miden a nivel local, es decir, cómo de importante es el nodo con respecto a los vecinos, y las que miden a nivel global permitiendo saber cómo de importante es el nodo respecto a la estructura completa de la red.

Antes de comenzar con el estudio de las métricas, cabe hacer una pequeña aclaración inicial en cuestiones de nomenclatura. Se considera que el grafo G(N, E) es un conjunto de nodos (N) enlazados entre sí mediante un conjunto de aristas (E). Los enlaces se representarán mediante una matriz de adyacencia (A) donde el elemento a_{ij} representa el enlace entre el nodo *i* y el nodo *j*. El resto de las cuestiones de nomenclatura se aclararán a medida que vayan apareciendo.

Grado de centralidad

Se comenzó por valorar a la que probablemente sea la medida más popular, dentro de las consideradas de centralidad local, además de las más simples, y no por ello poco útil. Si definimos el grado de centralidad de un nodo (C_i), dentro de un grafo no dirigido, como el número de enlaces que lo conectan con otros nodos [9]:

$$C_i = \sum_j a_{ij}$$

Siendo a_{ij} los elementos correspondientes a la matriz de adyacencia (A). Esta medida, aparte de necesaria para el cálculo de otras medidas, se puede interpretar como una cuantificación de la influencia e importancia que tiene en términos absolutos, es decir, los nodos con más enlaces tienen una mayor centralidad y, por tanto, puede llegar a más nodos [10]. No obstante, esta medida no habla del todo de la importancia real del nodo en el global de la red, sino únicamente sobre sus vecinos.

Intermediación (betweenness centrality)

Además de la importancia que puedan tener de por sí el cálculo de los caminos geodésicos, esto también sirven para obtener la intermediación de los nodos, otra medida de centralidad local. Definimos la intermediación (C_i^{Bet}) de un nodo *i* como [11]:

$$C_i^{Bet} = \sum_{j,k} \frac{b_{jik}}{b_{jk}}$$

Siendo b_{jk} el número de caminos geodésicos entre los nodos *j* y *k*, y siendo b_{jik} el número de caminos geodésicos entre los nodos *j* y *k* que pasen por el nodo *i*. Se obtiene con esta medida cómo de importante es la posición de un nodo respecto a la estructura total de la red. Es decir, si su posición es realmente crítica para la red entera significa que estamos ante un nodo con una gran influencia dentro de la red.

Cercanía (closeness)

Otra de las medidas más populares para el cálculo de la centralidad local de un nodo es la cercanía. Lo que se busca con esta métrica es calcular como de accesible es un nodo a la red. Si definimos *d(i,j)* como la distancia geodésica entre los nodos *i j*, podemos definir la cercanía (C_i^C) de un nodo *i* como:

$$C_i^C = 1 \Big/ \sum_{j=1}^{g} d(i,j)$$

La interpretación de esta métrica es la capacidad que tiene un nodo de llegar al resto de nodos, es decir, lo cercano que está de la mayoría de nodos.

Centralidad de vector propio (eigenvector centrality)

La centralidad de vector propio es una medida de centralidad global y algo más compleja que las vistas hasta ahora, además de más completa a la hora de estimar la centralidad dentro de las redes ponderadas, es decir, donde los pesos de los enlaces cuentan. Es considerada una métrica de centralidad global, en concreto, esta métrica calcula la importancia de un nodo en base a la centralidad de sus vecinos, es decir, no basta con que el nodo tenga muchos vecinos como puede ser el caso del grado, sino que se pondera cómo de centrales son los vecinos.

La centralidad de vector propio de un nodo se define como la suma de las centralidades de los nodos conectados a él [12]. Siendo λ el mayor valor de centralidad de vector propio de A y siendo *n* el número de nodos:

$$Ax = \lambda x, \quad \lambda x_i = \sum_{j=1}^{n} a_{ij}\, x_j$$

Además de esta medida de centralidad de vector propio, existen otras formas distintas, siendo una de las más extendidas la propuesta por Bonanich que se define como:

$$c_i(\alpha, \beta) = \sum_j (\alpha + \beta c_j)\, A_{ij}$$

La principal bondad de esta medida es que incluye parámetros que permiten adaptar el cálculo, siendo α un parámetro que únicamente afecta a la longitud del vector, es decir, un parámetro de normalización [13], mientras que β es un parámetro de atenuación que permite determinar cuánto influyen los enlaces más lejanos, ajustando el cálculo a las necesidades y a la estructura de la red. Por ejemplo, si el valor de β es grande se tendrá en cuenta el peso de la estructura global de la red, en cambio con un valor bajo tendrán mayor peso los vecinos cercanos.

Page Rank

Es conocido por ser el algoritmo que catapultó a la fama a sus creadores, Brin y Page, al implementarlo en su buscador web Google y convertirlo en el mayor buscador de Internet hoy en día. Esta métrica está considerada también como una medida de centralidad global. Es una variante de la centralidad de vector propio puesto que también valora la importancia de un nodo en base a la importancia de los nodos vecinos. La idea principal es poder crear un ranking de las páginas web aprovechando que la estructura de la web puede representarse como un grafo de páginas que apuntan a otras páginas.

Para definirlo de una forma sencilla, se designa *u* como una página web. Consideramos que F_u es el conjunto de páginas a las que apunta *u*, B_u el conjunto de páginas que apuntan a *u* y $N_u = |F_u|$ el número de links desde *u* y *c*, que será utilizado como un factor de normalización. Con esto obtenemos que el ranking se puede definir [14]:

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v}$$

Sin entrar a profundizar más en el algoritmo, que puede consultarse en el trabajo original [14], se observa como la importancia recursiva de los vecinos influye en el ranking del propio nodo en un estilo similar a la centralidad de vector propio, permitiendo obtener un valor de la importancia dentro de la totalidad de la red.

HITS

El algoritmo HITS (Hypertext Induced Topic Selection), diseñado por Kleinberg [15], forma parte también de las medidas de centralidad global, siguiendo la misma línea que PageRank puesto que considera la idea de la web como un enorme grafo de referencias entre páginas y pretende clasificar la importancia de cada una. El funcionamiento del algoritmo se puede definir con dos conceptos sencillos *hubs y authority*.

Lo que propone Kleinberg es valorar una página con dos valores recursivos entre sí, su valor como *hub* determinará la calidad de la información que se obtiene siguiendo los enlaces que tiene una página web (las páginas a las que apunta). Por el otro lado, el valor cómo *authority* determinará cómo de buena es la información contenida en la página en base al número de *hubs*

que la apuntan. Simplificando, un buen *hub* es aquel que apunta a páginas con un alto valor de *authority* e, inversamente, una página será buena *authority* si es apuntada por páginas con un nivel alto de *hub*. Cómo se puede observar, este algoritmo sigue la idea de la centralidad recursiva, es decir, un nodo tiene una mayor centralidad si sus vecinos tienen una alta centralidad.

Coeficiente Clustering

El coeficiente local de clustering es una de las medidas de cohesión de un nodo, es decir, mide como de unidos están los vecinos de un nodo. Si un nodo *i* tiene $n_i$vecinos y tenemos que L_i es el número de enlaces directos entre los vecinos del nodo, se define:

$$C_i^{CLC} = \frac{2L_i}{n_i(n_i - 1)}$$

En el caso de que el coeficiente sea 0 significa que sus vecinos no están conectados entre sí y, en el caso opuesto, de que sea 1 significará que todos los vecinos se encuentran conectados. Esta métrica resulta interesante para medir la transitividad de la red, algo que en las redes sociales toma especial relevancia respecto a otros modelos de redes [16], sirviendo para poder comprender la cohesión existente entre un nodo y sus vecinos pudiendo identificar comunidades y complementando los valores obtenidos por las otras métricas.

Además, dada la relación, se decide incorporar a la lista de métricas el *coeficiente de clustering global* que, a diferencia de las otras medidas, se encuentra entre las medidas globales de cohesión de una red y permite cuantificar como de unida está la red en su conjunto global.

2.4 Visualización de redes

Dentro de la teoría de grafos existe todo un subcampo dedicado única y exclusivamente a la representación y visualización del grafo. Lo que permite la visualización de un grafo es captar la estructura de la red que lo conforma de una forma gráfica e intuitiva, mucho más fácil de interpretar que la representación matricial de un grafo.

A priori, se podrá considerar la representación de un grafo como algo trivial, consistente en ubicar los nodos en un plano y unirlos mediante las aristas. Sin embargo, existen una serie de problemas derivados de la propia naturaleza del grafo pudiendo ser el principal el tamaño del mismo. Si el grafo es excesivamente grande podemos encontrar limitaciones de espacio y/o de

rendimiento de la propia plataforma de visualización. Además, se generan también problemas de legibilidad y usabilidad ya que puede llegar un punto en el que los nodos y enlaces se superpongan no permitiendo discernir cual es cual y, por tanto, imposibilitando la comprensión de la red [17].

Para evitar este tipo de problemas es necesario seguir una serie de principios estéticos que permitan facilitar la visualización de la red:

- Minimizar los cruces entre enlaces
- No permitir que los nodos se superpongan a enlaces
- En redes ponderadas, hacer que la longitud del enlace sea proporcional a su peso

Para poder cumplir con estos principios existen diferentes *layouts* o algoritmos de distribución para representar las redes. A continuación, se nombrarán los principales layouts comentados por su función, sus virtudes y desventajas.

Layout aleatorio

Su funcionamiento es simple y persigue la idea más ingenua de la visualización de los grafos ya que consiste en ubicar los nodos de manera aleatoria dentro del plano y unirlos mediante los enlaces. Si bien tiene cómo ventaja la rapidez de cálculo, tiene cómo desventajas que no muestra la estructura real de la red y, además, incumple todos los principios estéticos de la representación de redes.

Layout de árbol

Una de las distribuciones más clásicas de un grafo consistente en ubicar a los nodos debajo su antecesor común, dando una representación fiable para redes con estructuras jerárquicas. Además, tiene cómo ventaja un buen rendimiento, especialmente con las redes que de por sí presentan una estructura con una marcada jerarquía, siendo al mismo tiempo su principal limitación [17].

Layout radial

Esta distribución, derivada del layout de árbol, consiste en ubicar al nodo central en el plano y distribuir el resto de los nodos en capas concéntricas alrededor de este en base al peso de sus enlaces. Cómo ventaja tiene un buen rendimiento de cálculo. Sin embargo, no considera los

enlaces entre los otros nodos para la ubicación por capas, pudiendo dar lugar a un gran número de cruces de enlaces y no respetar la longitud uniforme en base al peso de los enlaces en las redes ponderadas.

Layouts guiados por fuerzas

Si bien los layouts guiados por fuerzas no son en sí mismos un tipo de distribución, sí engloban una serie de algoritmos que basan la representación de los grafos en un modelo de objetos unidos por fuerzas. La idea es representar de la forma más fiel posible la unión de la red, suponiendo que los enlaces generan una fuerza de atracción entre los nodos, de forma que cuanto mayor sea el peso del enlace más cercanos estarán los nodos y, por el contrario, si el enlace es muy débil la distancia será mayor. Este tipo de representaciones permite obtener una visualización que respeta, en la medida de lo posible, los criterios estéticos, aunque a cambio tienen un coste de cálculo muy superior a las distribuciones básicas vistas. Buenos ejemplos de este tipo de representaciones son los algoritmos propuestos por Fruchterman & Reingold [18] o por Yifan Hu [19].

2.5 Conclusiones

Wikia es, como se ha visto, un gran host de comunidades de distintos tamaños y, previsiblemente, distintos patrones de funcionamiento, siendo el lugar idóneo donde estudiar el comportamiento de los usuarios dentro de una wiki. Dado el tamaño de Wikia se puede explorar la influencia de diversos factores, como pueden ser el tamaño de la wiki o la madurez de esta, en la conducta de los usuarios. Entendemos ese comportamiento como su manera de interactuar con la comunidad, por ejemplo, en el reparto de la carga de trabajo que supone mantener la wiki y generar nuevo contenido. Si suponemos que la conducta de los usuarios puede influir en el devenir de la wiki y que mediante el estudio se pueden llegar a determinar patrones de comportamiento, se puede obtener no solo un material de gran valor sociológico, sino también modelos predictivos en el funcionamiento de una wiki.

Además, debe añadirse a esta ayuda la madurez del análisis de redes como campo de estudio, proporcionando material extenso que permite determinar cuáles son las métricas de mayor relevancia. Como se ha podido ver, esto ha ayudado para lograr desarrollar una herramienta de gran utilidad y con suficiente versatilidad, dando también las técnicas suficientes para visualizar

una red de forma óptima para simplificar la información de la misma y transmitirla de manera gráfica pudiendo comprender la estructura de la red.

Teniendo en cuenta lo anterior, las posibilidades que brinda la herramienta que propone el presente trabajo en conjunción con otros proyectos, puede suponer todo un avance para estudio de las comunidades de producción colaborativa, permitiendo profundizar a nivel sociológico en un modelo que es, al menos en el plano teórico, bandera de una forma más equitativa y cooperativa de avanzar hacia un bien común.

Capítulo 3 - Diseño de la herramienta

Una vez explicado qué es y cómo funciona Wikia, y las bondades del análisis de redes queda por definir cómo puede servir esto para el diseño de la herramienta y como se pretende abordar el desarrollo de la misma.

Tal y como se comentó con anterioridad, el objetivo de la aplicación es poder extraer información de una wiki y generar un grafo asociado a la misma con una serie de métricas que nos puedan aportar cierta información sobre la wiki. Además, entre las características de la herramienta debería encontrarse la facilidad de uso y la versatilidad de poder emplearlo en el mayor número de plataformas posibles. Por tanto, el funcionamiento de la herramienta podría dividirse en tres grandes fases compuestas por la extracción de los datos de la wiki, la generación del grafo asociado a la misma y su análisis y, finalmente, la visualización y muestra de los resultados al usuario.

3.1 Prototipo de diseño

La idea es crear una aplicación web similar a la que se ve en el boceto de la Figura 3.1, en donde el usuario pueda elegir una wiki a estudiar, entre las existentes en Wikia, y obtenga la representación de los usuarios que participan en la misma y como se relacionan entre ellos. Dicha representación será un grafo en el que cada nodo represente a un usuario y las aristas que los unen las páginas editadas en común o, el grafo contrario, en el que los nodos serían las páginas y los enlaces los usuarios.

Además, se añadirá la posibilidad de elegir distintas métricas que permitan determinar cómo de centrales e importantes son cada nodo o cómo de cohesionada es la red. Esto puede ser útil de cara al estudio de comportamientos dentro de redes de producción colaborativa ya que, si en una wiki solo unos pocos nodos tienen gran importancia, significa que la carga trabajo no se está repartiendo de forma equitativa.

Estas métricas, junto con otros valores, modificarán las dimensiones visuales del grafo (tales como tamaños de los nodos, grosor de las aristas, etc.) con el fin de facilitar la compresión

de este de forma gráfica. Siguiendo con la parte de representación, se permitirá elegir al usuario entre distintos layouts de representación que permitan captar la estructura real de la comunidad.

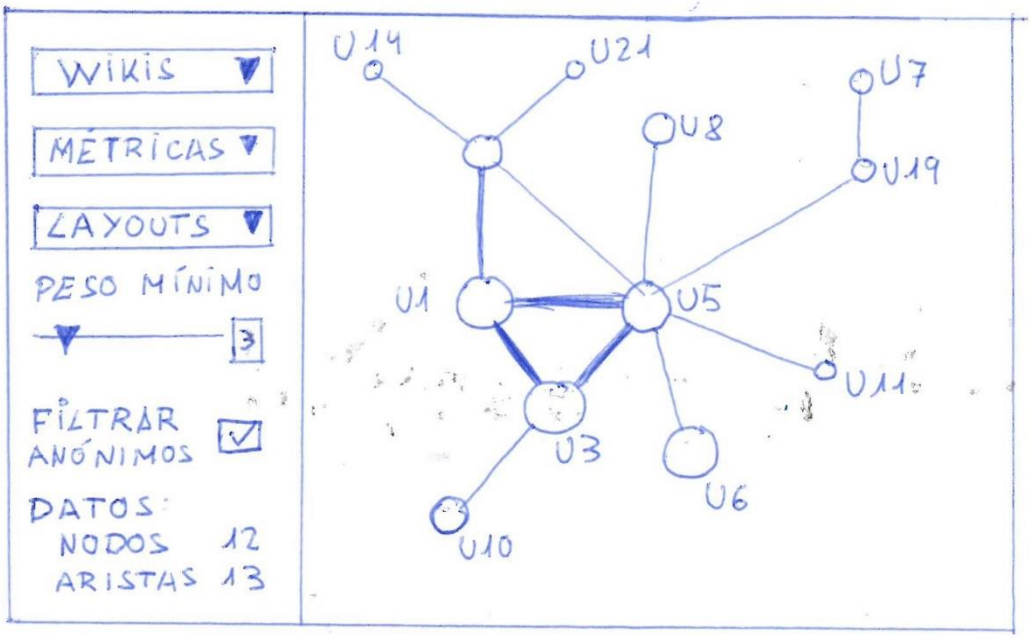

Figura 3.1 Boceto de la aplicación

También, se pretende implementar un filtrado de nodos en base a si son usuarios registrados o anónimos que, tal y como vimos en el anterior capítulo, pueden estar presentes en los registros de una wiki y acabar generando ruido en el estudio de esta. Por último, se implementará también un filtrado de las aristas en base a su peso, es decir, aquellas aristas con un peso inferior al elegido serán eliminadas de la representación. Esta es una práctica muy común dentro de la visualización de grafos, pues permite eliminar los enlaces que, por su peso, se consideren despreciables al mismo tiempo que limpia la representación facilitando su estudio. Cabe decir que junto con este filtrado de aristas se pretende implementar una limpieza de nodos aislados que lo complemente. Por ejemplo, si nos encontramos ante el estudio de una wiki de gran tamaño y decidimos considerar que los enlaces de peso inferior a tres no suponen datos de interés para nuestro estudio. Mediante el filtrado obtendremos una visualización más clara de la comunidad, dado que se pueden eliminar una cantidad de enlaces y nodos cuyo aporte a la wiki sea desdeñable y nos permita entonces identificar con mayor facilidad los usuarios más importantes de esta.

Como ya se comentó con anterioridad, el funcionamiento de la herramienta se podría dividir en tres fases principales: extracción, análisis y representación. Para la primera, y debido a que la herramienta está prototipada para funcionar con Wikia, se deberá extraer los datos mediante la API proporcionada por esta y convertirlos a un formato fácilmente manipulable. Una vez extraídos, llegaría la segunda fase donde se analizarán los datos y se generará el grafo asociado junto con sus métricas. Finalmente, se deberá mostrar al usuario una representación del grafo y las métricas obtenidas en la fase anterior. A continuación, se explicará con mayor detalle cada una de estas fases.

3.2 Extracción de datos

Siguiendo la secuencialidad de funcionamiento, el primer paso a realizar para el desarrollo de la herramienta debe ser obtener los datos de una wiki para poder analizarlos. Obviamente, la extracción manual de los datos es inviable y ni siquiera se contempla, pero para eso Wikia pone a nuestra disposición una API bastante versátil, pero con algunas limitaciones.

Entre las herramientas creadas por los compañeros de la facultad[10], se dispone de un script que, dada la *url* de una wiki, descarga un "dump" o registro con todos los cambios realizados en esa wiki, es decir, devuelve un histórico de los cambios que se han realizado. Sin embargo, el archivo devuelto tiene dos inconvenientes: el primero es que viene en formato xml que, si bien es un formato idóneo y extendido para el envío por la red, no lo es tanto para el tratamiento masivo de datos. El otro problema es que trae demasiada información que puede no ser de utilidad como, por ejemplo, el texto con el cambio integro que se realizó en cada revisión, algo que aumenta notablemente el tamaño del archivo y no proporciona información que, a priori, pueda ser útil.

Además, cabe destacar la limitación de la propia API que no envía dumps de más de 5000 páginas. Por tanto, en caso de que la wiki en cuestión sea relativamente grande, será necesario fraccionar el dumps en archivos de 5000 páginas. Sin embargo, para salvar estas limitaciones, los compañeros de la facultad desarrollaron dos scripts, uno para juntar todas las partes en un único archivo y otro script para procesar los datos descargados y limpiarlos generando como resultado

[10] https://github.com/Grasia/wiki-scripts

un archivo en formato CSV en el que las filas son las revisiones que se han realizado en la wiki y cada columna corresponde a los siguientes datos de cada revisión:

- **Page_id:** El ID de la página que ha sido modificada.
- **Page_title:** El título de la página en cuestión.
- **Page_ns:** Es el namespace de la página[11].
- **Revision_id:** El ID de la revisión.
- **Timestamp:** Fecha y hora a la que se realizó la revisión.
- **Contributor_id:** El ID del usuario que realizó la revisión. En caso de ser un usuario anónimo será la IP desde donde se realizó el cambio.
- **Contributor_name:** El nombre de usuario. En caso de ser anónimo vendrá el valor "Anonymous".
- **Bytes:** El tamaño, en bytes, de los cambios realizados en la revisión.

Con esto ya contamos con los datos de una wiki en un formato de fácil trabajo y con la información necesaria para poder generar grafos que permitan estudiar el comportamiento de la wiki.

3.3 Análisis de los datos

Una vez conocidos los datos de los que se disponen, es necesario definir cómo se van a tratar para obtener el resultado deseado. Dado que lo que se pretende es poder visualizar y analizar el comportamiento y la interacción de los usuarios dentro de una wiki, una de las formas óptimas de hacerlo es generando un grafo a partir de los datos disponibles que representen la red de la wiki.

Si observamos los datos disponibles, vemos que tenemos una relación entre usuarios y páginas que podría describirse como un usuario u_i que ha editado una página p_i, identificando al usuario y a la página de manera inequívoca con su ID (a excepción de los usuarios anónimos que serán tratados de forma especial) y a las páginas también con su ID. Es decir, el resultado de esta primera transformación se podría representar como grafo bipartito, en el que uno de los conjuntos

[11] http://community.wikia.com/wiki/Help:Namespace

de nodos sean los usuarios y los nodos del otro conjunto las páginas y las aristas representarían que un usuario ha editado una página.

En un grafo bipartito tenemos los nodos separados en dos conjuntos diferenciados y los nodos se relacionan únicamente con nodos que no son de su conjunto. En nuestro caso tenemos usuarios que podemos relacionar con páginas, pero no existe una relación directa entre usuarios o entre páginas. Para conseguir esa relación necesitamos transformar un grafo bipartito en un grafo no bipartito mediante la proyección del grafo [6]. En la Figura 3.2 se ilustra un ejemplo de proyección de un grafo bipartito.

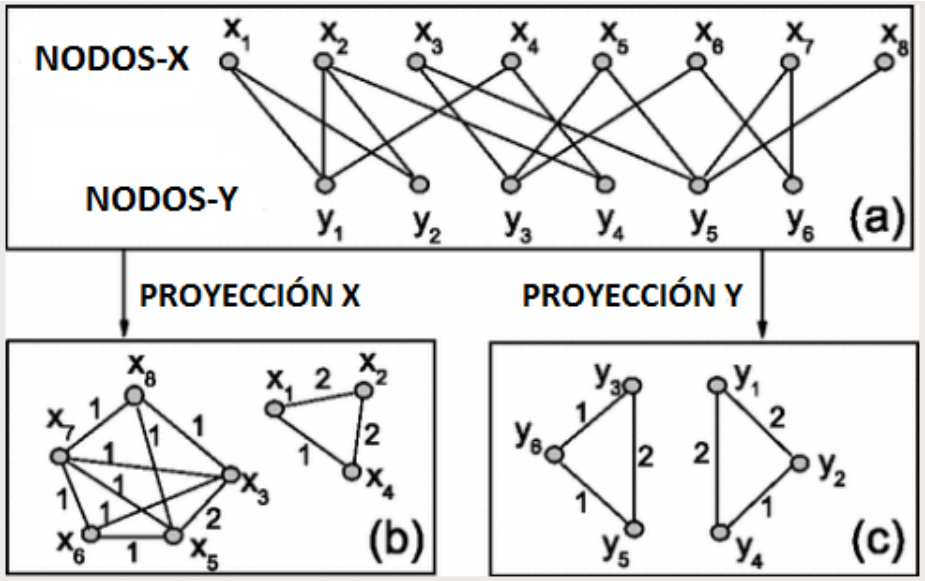

Figura 3.2 Proyección de un grafo bipartito [6]

No obstante, la proyección de un grafo bipartito genera dos grafos distintos. En el caso de este trabajo, se puede generar un grafo en el que los nodos sean los usuarios y las aristas sean las páginas comunes que han editado ambos o, el caso inverso, en el que los nodos sean las páginas y las aristas representan los usuarios comunes que hayan editado a ambas. Como el fin del presente trabajo es el de generar una herramienta lo más versátil posible y dado que no se puede llegar a la conclusión de cuál de las dos proyecciones resultará más útil para el estudio de la wiki, se decide,

por conveniencia, que la herramienta soporte y trabaje con ambas proyecciones, quedando en manos del usuario final la elección.

En lo respectivo al análisis del grafo resultante disponemos de una gran batería de métricas a aplicar, cómo se vio en el capítulo anterior.

- **Grado de centralidad**: Se consideró necesario incluir esta medida dentro de la herramienta, tanto por su popularidad como por sus posibilidades de interpretación y por su facilidad de representación
- **Intermediación**: Interpolando al caso del proyecto, nos diría cómo de importante es el nodo dentro de la comunidad y en qué medida el funcionamiento de la misma pueda depender de él.
- **Cercanía**: Llevado el valor de esta medida a nuestro proyecto podría suponer que un nodo con una cercanía alta sea un nodo de gran importancia dentro de la comunidad por su capacidad para interaccionar con muchos nodos además de generar una gran cohesión dentro de la misma.
- **Centralidad de vector propio**: Con esta medida conseguimos calcular la centralidad y ajustarla en base a un nivel de localidad del nodo, por tanto, podremos determinar la centralidad de un nodo y observar su importancia tanto local como global variando el valor de β.
- **Page Rank**: En nuestro caso, resulta útil para identificar a los usuarios más influyentes y de mayor peso dentro de la comunidad, gracias a que valora la centralidad de este de forma global.
- **Coeficiente de clustering**: Esta métrica permite calcular la cohesión de la red de forma sencilla tanto a nivel local como global, dando una idea de cómo de unida esta la comunidad. Dicho de otra forma, cuanto más cohesionada este la red mayor será la interacción de los usuarios entre sí y, por tanto, habrá una distribución del trabajo más equilibrada.

Por otro lado, en lo referente a los pesos de los enlaces, aunque existen diversas formas de aplicarlos, se consideró que en el presente trabajo y con los datos proporcionados no tendrían especial interés estudiar técnicas elaboradas de ponderación. Por tanto, como única medida, se les

asignará a los enlaces un peso de una unidad por cada nodo en común que tengan en el grafo bipartito dos nodos de un mismo conjunto. Llevándolo a un ejemplo práctico, si tomamos el grafo en el que los usuarios son los nodos y las páginas editadas en común los enlaces, dos nodos tendrán un enlace de peso 1 en el caso de que solamente hayan editado una misma página; en cambio, en el caso de que tengan en común la edición de 5 páginas distintas, tendrán un enlace con peso 5. Con esto se consigue poder cuantificar la fortaleza de un enlace, es decir, como de cercana es la interacción entre los usuarios. Además, el peso de los enlaces influye en el cálculo de algunas medidas de centralidad de los nodos, dando resultados más precisos y cercanos a la realidad.

3.4 Representación del grafo

Por último, quedaría por especificar cómo será la visualización del grafo. Tal y como se vio en el capítulo anterior, existen diferentes layouts o algoritmos de distribución. Sin embargo, de todos los vistos serían los layouts guiados por fuerzas los que mejor se adaptarían a las necesidades del proyecto. Esto nos llevaría a descartar la distribución aleatoria ya que una comunidad como es una wiki siempre llevará una cierta organización dentro de sus relaciones, además que no permitiría apreciar la estructura real de la red. En lo referente a los layouts de árbol y radiales, funcionarían bien si la red fuera fuertemente jerarquizada, algo que contravendría con la propia naturaleza de la wiki donde se suponen comunidades poco jerarquizadas. Es por esto por lo que todos los algoritmos de distribución guiados por fuerzas pueden servir para visualizar de forma óptima el grafo, ya que, entre otras cosas, respeta los pesos de las aristas en la representación, permitiendo obtener una idea más precisa de la verdadera estructura de la red.

Además del layout de distribución existen otros elementos de la representación que pueden modificarse. Estos son las dimensiones visuales de los distintos elementos que componen el grafo. En el caso de los nodos nos encontramos que las dimensiones visuales más relevantes son el tamaño, la forma, el color y la etiqueta. En el caso de este proyecto, puede resultar útil la etiqueta del nodo para mostrar a quién representa, es decir, en uno de los grafos resultantes la etiqueta contendría el nombre del usuario y en el otro grafo resultante podría contener el título de la página que representa. También, tanto el color como el tamaño suelen ser dimensiones recurrentes para mostrar el valor de una métrica de un nodo, por ejemplo, dando mayor tamaño a los nodos con

mayor valor de una cierta métrica (e inversamente menor tamaño a menor valor) o asignando un color al valor más bajo y otro al más alto y que cada nodo adquiera el tono dentro de la escala marcada en base a su métrica. Sin embargo, como el caso del color puede resultar a veces confuso, para el presente proyecto se opta por asignar un mayor tamaño a los nodos en base a una métrica deseada.

En lo referente a las aristas, sus dimensiones visuales son, entre otras, el color, grosor, su opacidad, su patrón o trama, o su etiqueta. En este caso, se consideró que añadir etiquetas a las aristas no suponía una ganancia de información y si un aumento del ruido visual, por tanto, se decidió no incluirlas. Además, al igual que sucede con los nodos, es habitual variar alguna de las dimensiones de las aristas en base a su peso y, aunque el layout ya se encarga de modificar su longitud, puede ser interesante para remarcar su peso variar el grosor, es decir, a mayor peso mayor ancho para la arista.

Capítulo 4 - Desarrollo

Para el desarrollo de la herramienta se optó por un modelo incremental, en el que se fuesen creando de manera prioritaria las partes más fundamentales para el funcionamiento de la aplicación e ir aumentando las funcionalidades hasta obtener el resultado final. El código resultante se puede ver en este repositorio (https://github.com/david-FV/TFM-Wikia-Tool).

Las fases podrían separarse principalmente en tres, siendo la primera el tratamiento de los datos y, a partir de ahí, la generación del grafo junto con el cálculo de las métricas necesarias y su exportación. La siguiente fase consistirá en estudiar y valorar distintas herramientas para la visualización y correcta representación del grafo y las métricas calculadas, a ser posible, en un formato dinámico y que permita la interacción con el usuario. Cómo última fase se encuentra el estudio y uso de diversas tecnologías web que permitan unir los módulos creados en las anteriores fases bajo una estructura de aplicación web que facilite el uso de la herramienta.

4.1 Procesamiento de datos y generación del grafo

Cómo se explicó en el capítulo 3, los datos de los que se parte son el dump de una wiki, con todo el historial de los cambios registrados que, previa limpieza, acaba resultando un archivo CSV donde cada fila es una revisión y cada columna información sobre la misma (usuario, página, fecha, etc.). Para comenzar, el primer objetivo será poder manejar los datos de forma programática.

Antes de esto será necesario definir en qué lenguaje de programación se desarrollará el núcleo de la aplicación, siendo la elección final Python. Existen múltiples motivos que llevaron a la elección de este lenguaje por encima de otros. La primera razón es su versatilidad, ya que, al ser extremadamente popular y extendido, cuenta con una buena colección de librerías de calidad para la realización de tareas de diversa índole, siendo de especial notoriedad las relacionadas con el tratamiento de datos, el manejo de grafos y cálculos matemáticos.

Además, se trata de un lenguaje multiparadigma y multiplataforma que permite su ejecución en distintos tipos de máquina. Otra de las razones de peso es la herramienta IPython[13], o su hermana mayor, Jupyter Notebook[14], que permiten ejecutar código Python de forma interactiva, pudiendo ver el resultado de la ejecución de cada línea o de pequeñas partes de código. Esto facilita enormemente el análisis, puesto que se pueden realizar pruebas del código de forma modular y exportarlo de forma sencilla para integrarlo en la aplicación. También hay que considerar como razón de peso el conocimiento del lenguaje por parte del autor y desarrollador del TFM.

Igualmente es destacable que Python es utilizado cómo lenguaje de *scripting*, algo realmente importante para el comienzo del desarrollo si se tiene en cuenta que el primer paso será crear un script que, dado un dump, pueda extraer los datos y generar un grafo.

El primer paso sería el manejo del archivo CSV y la consecuente extracción de los datos algo que se logra fácilmente con las librerías nativas de Python. El siguiente paso será generar el grafo asociado. Tal y cómo se explicó en el capítulo 3, de los datos disponibles podemos obtener con facilidad un grafo bipartito. El problema es que este grafo de por sí no resulta útil para mostrar cómo se relacionan los miembros de un mismo conjunto, es decir, los usuarios entre sí o las páginas entre sí. Para ellos, tenemos que realizar una proyección del grafo bipartito generando dos grafos distintos.

Para facilitar la lectura se explicará cómo se hizo la proyección y la generación del grafo para el caso en el que los usuarios son los nodos de la red y las páginas los enlaces. Así, si se quisiera entender cómo generar el otro grafo, bastaría con invertir los valores entre usuarios y páginas, puesto que la lógica es la misma.

[13] https://ipython.org/

[14] http://jupyter.org/

Para la generación y tratamiento del grafo se decidió utilizar la librería de Python *NetworkX*[15], una librería extensamente utilizada en el tratamiento de grafos, con un amplio abanico de funciones, un buen rendimiento con redes de casi cualquier tamaño y una madurez como proyecto que la convierten en una herramienta de enorme valía y facilidad de uso.

Con las funciones de la librería tendremos suficiente para crear el grafo a partir de los datos disponibles. Para generar la proyección, iteraremos por todo el archivo obteniendo de cada revisión el id de la página modificada y el id del usuario que la modificó, almacenando su valor en un diccionario donde las claves serán los id de las páginas y el valor un array con los identificadores de todos los usuarios que hayan modificado dicha página.

Un ejemplo de diccionario resultante sería:

```
{
  "P1": ["U1", "U5"],
  "P2": ["U2", "U3", "U1"],
  "P3": ["U1", "U2"]
}
```

Siendo *Pi* el ID de la página *i* y *Ui* el ID del usuario *i*. Además, en versiones posteriores y por cuestiones de legibilidad se decidió almacenar también el nombre de usuario para añadirlo a los atributos de los nodos a modo de etiqueta. Una vez obtenido el diccionario, bastará con iterar sobre él, añadiendo al grafo como nodos los usuarios almacenados en el array sin olvidarse de comprobar que el nodo no exista previamente para evitar sobrescribir la información existente. Además, añadiremos los enlaces necesarios para unir todos los nodos contenidos en un mismo array, asignándole un peso de valor 1 en caso de que el enlace no existiera previamente o, en caso contrario, incrementando el valor del peso en 1.

Siguiendo con el ejemplo anterior y por ejemplificar, en la primera iteración se añadirían al grafo los nodos asociados a los usuarios *U1* y *U5*, además de crearse el enlace que los una con un peso de valor 1. En la segunda iteración se añadirían los nodos de los usuarios *U2* y *U3*, puesto

[15] https://networkx.github.io/

que *U1* ya fue añadido en la iteración anterior. También se crearían los enlaces que unieran a *U2* con *U3 y* con *U1* y otro más que uniera a *U3 y U1*, todos de peso 1, puesto que son de nueva creación. Finalmente, en la última iteración no sería necesario añadir ningún nodo ni enlace nuevo, sino que se incrementaría en 1 el valor del enlace que une a *U1 y U2*.

Una vez que el grafo fue generado se puede empezar a explotar todo el potencial que nos brinda NetworkX, empezando por las funciones necesarias para calcular todas las métricas que se habían propuesto en el capítulo 2. Dado que en esta primera fase de desarrollo no se ha implementado aún manera de visualizar el grafo ni las métricas calculadas, se toma como opción para comprobar el buen funcionamiento del script volcar el resultado de cada una de las métricas en un fichero CSV.

Llegados al final de la primera fase de desarrollo, se tiene como resultado un script que dado un dump permite generar el grafo asociado (tanto de usuarios como de páginas) y calcular una serie de métricas que devuelve en un fichero CSV. Sin embargo, salvo pruebas modulares realizadas en IPython, no se tiene una manera fiable de saber que el resultado generado este yendo por buen cauce, ya que no se tiene ninguna visualización del mismo.

Para solucionar este problema y poder dar por concluida la primera fase de desarrollo se opta por exportar el grafo generado a un formato aceptado por alguna aplicación de tratamiento y análisis de grafos. En concreto, se va a utilizar Gephi[16], una aplicación de software libre, gratuita y especialmente creada para el manejo, visualización y análisis de grafos, convirtiéndola en una candidata ideal para las funciones requeridas.

El siguiente paso para poder visualizar el grafo, será conseguir exportarlo a algún formato aceptado por Gephi. Para lograrlo, la librería NetworkX ofrece una serie de funciones que permiten exportar el objeto de tipo grafo a distintos formatos de representación como, por ejemplo, graphml.

[16] https://gephi.org/

Basta con pasar como parámetros el grafo y la ruta del archivo donde se quiera exportar que la librería se encargará de realizar la conversión. Además, provee opciones de configuración como elegir la codificación del resultado o añadir tabulación que facilite la lectura del XML resultante. En la misma línea nos encontramos con funciones para exportar el grafo en formato GEXF, GML o Pajek entre otros. Dada la facilidad de implementación y la popularidad de los mismos se decide añadir al script la funcionalidad de exportar el grafo en los cuatro formatos tratados.

Ya con la funcionalidad implementada, se pudo exportar el grafo a un formato aceptado por Gephi y visualizarlo en la aplicación. Gracias a esto se pudo comprobar que la primera fase del desarrollo se había cumplido con éxito, permitiendo observar los grafos asociados a múltiples wikis. Además, dentro de las pruebas iniciales de visualización, se observó que el grafo generado realizando la proyección en la que los nodos de la red son las páginas y los enlaces los usuarios, tiene, incluso en las wikis más pequeñas, unos tamaños muy grandes para su manejo y visualización. Esto genera que en el caso de wikis medianas, grandes y muy grandes el grafo asociado a esta proyección tenga un uso y trabajo limitado. Sin embargo, como no supone ningún coste extra en el desarrollo del proyecto y es un valor añadido para la herramienta, no se descarta su implementación en la aplicación final.

También, otra observación que se obtuvo de las primeras pruebas con Gephi fue la cantidad de ruido visual que generaban las etiquetas de los nodos, especialmente en las redes grandes y teniendo en cuenta que los nodos anónimos tienen siempre el mismo nombre. Para solventarlo y, de paso, incrementar la funcionalidad del script se modificaron las funciones de generación del grafo para poder elegir si se desea eliminar los usuarios anónimos o, en caso de mantenerlos, no asignarles etiqueta a esos nodos, facilitando su identificación y limpiando la visualización del grafo. Por último, y siguiendo la línea de facilitar la visualización del grafo, se añadió la posibilidad de elegir un peso mínimo de las aristas, eliminando aquellas que se encuentren por debajo del valor elegido y, como complemento a esta funcionalidad, se eliminarán del grafo los nodos aislados.

4.2 Visualización del grafo

La segunda fase del desarrollo consiste en la implementación de un método de visualización dinámico del grafo generado por el script. Además, otro de los requisitos es que pueda ser fácilmente integrado como aplicación web.

Por un lado, se estudió la posibilidad de utilizar alguna de las posibilidades de dibujo y visualización que ofrece NetworkX. Sin embargo, esta opción resultó no ser viable, ya que las opciones de dibujo ofrecidas no generan una visualización dinámica que permita al usuario interactuar con ella, limitando notablemente el uso de la herramienta. Además, los propios desarrolladores de NetworkX avisan en la documentación que esta opción está obsoleta y se eliminará en futuras versiones, invitando a usar aplicaciones de visualización como Gephi, Cytoscape o GraphViz.

Otra opción podría ser la de utilizar, como se hizo en la primera fase, Gephi para visualizar el grafo. No obstante, esta opción tiene dos grandes problemas, puesto que hace que el funcionamiento de la herramienta dependa de una aplicación externa y, además, complica la integración como aplicación web.

4.2.1 Librerías de visualización

Para encontrar una solución que pudiera satisfacer los requisitos se decidió investigar el uso de librerías JavaScript de visualización de grafos, cumpliendo fácilmente el requisito de integración con la web. Tras una búsqueda sobre la oferta de librerías de este tipo, se hizo una selección final de tres posibles candidatas: D3, Sigma y Cytoscape.

D3.js

La librería D3.js[17] (Data-Driven Documents) está diseñada para facilitar la visualización de datos de distinta índole, siguiendo los estándares web. Funciona bajo una combinación de SVG, Canvas y HTML y permite la interacción con los gráficos generados. Además, proporciona compatibilidad con los principales navegadores web y cuenta con una API muy completa apoyada en una extensa documentación.

[17] https://d3js.org/

Sin duda alguna, D3.js es la librería más potente a nivel de posibilidades de las 3 analizadas. Sin embargo, tiene un principal problema. Esta librería no está pensada únicamente para la representación de grafos, sino de cualquier tipo de gráfico o representación de datos. Esto la convierte en una librería con un nivel de control sobre el gráfico excesivamente fino para el proyecto actual, ya que supondría a nivel de costes de desarrollo un esfuerzo extra con respecto a las otras dos librerías, sin tener que suponer una mejora en la calidad de la representación. Esta es la principal razón que lleva a descartar el uso de la librería D3.js.

Sigma.js

Sigma.js[18] es una librería de software libre, al igual que las otras dos, diseñada especialmente para el dibujo de grafos. En una línea de uso similar a Cytoscape.js, permite visualizar grafos pasados en un formato JSON propio, bastante sencillo de componer, y con la posibilidad de añadir funcionalidades mediante plugins.

Sin embargo, la API de Sigma.js es bastante más limitada que las APIs de las otras librerías, permitiendo menos opciones de personalización como, por ejemplo, mapear el tamaño de los nodos en base a un valor. Además, la documentación de la librería es bastante pobre en general y un tanto escasa en ejemplos de uso, algo que dificultaría el aprendizaje de manejo de esta. Por tanto, a la vista de que tiene características muy similares a Cytoscape.js, pero en ciertos puntos se ve superada por esta, se decide descartar su uso en la implementación de la herramienta.

Cytoscape.js

Cytoscape.js[19] (en adelante sólo Cytoscape) es una librería de código abierto pensada única y exclusivamente para el análisis y la visualización de grafos. Es desarrollada como un proyecto complementario de Cytoscape, una aplicación de escritorio de software libre similar al Gephi.

Cuenta con una más que aceptable API con multitud de opciones que viene acompañada de una holgada documentación y gran cantidad de ejemplos que facilitan el aprendizaje de uso, algo especialmente valorado si se tiene en cuenta la limitación de tiempo del proyecto. Además,

[18] http://sigmajs.org/

[19] http://js.cytoscape.org/

cuenta con una lista de *plugins* desarrollados por terceros para complementar las funciones de la librería. Permite entre otras cosas, la interacción del usuario con la representación, elegir diferentes layouts y modificar atributos de nodos y aristas.

La principal contra de esta librería es que utiliza un formato propio de grafo, lo que implica la necesidad de transformar el grafo generado por NetworkX. No obstante, el formato propuesto está en JSON y viene muy bien documentado, por lo que su implementación no supondría un coste elevado. Por estas razones, se decide elegir Cytoscape como librería para dibujar e interactuar con el grafo.

El uso de Cytoscape es bastante sencillo y gracias a una nutrida colección de ejemplos el aprendizaje es bastante rápido. El elemento base de la librería es el objeto Cytoscape que viene a ser un grafo con ciertas opciones de visualización. Los elementos mínimos que se deben pasar para la visualización son:

- **Container**: El elemento del HTML DOM donde se mostrará el grafo. Para su correcto funcionamiento se espera que sea un elemento div vacío.
- **Style**: El estilo a aplicar al grafo, es decir, cuestiones de visualización como puede ser el color y tamaño de los nodos y aristas, las etiquetas a mostrar de los nodos y/o aristas o, incluso, la tipografía elegida.
- **Elements**: Son los datos del propio grafo en un formato JSON propio. Es decir, aquí es donde se incluyen los nodos y aristas con sus atributos.
- **Layout**: El objeto que determinará la posición de los nodos dentro del campo de visualización. Si bien se puede especificar la posición de cada nodo de forma manual, se recomienda utilizar los objetos de layout existentes en la librería.

El resto de las opciones disponibles están relacionadas principalmente con cuestiones de interacción y renderizado del grafo. Como vemos, con solo definir cuatro opciones y hacer una llamada a una función tendríamos la visualización del grafo disponible, sin embargo, no todas las opciones tienen una elección trivial.

4.2.2 Dimensiones visuales

En lo referente al estilo de la visualización, el espectro es amplio y las opciones son variadas. Si bien Cytoscape tiene valores por defecto para la mayoría de las características, siempre es conveniente añadir un grado de personalización, además, modificar ciertas características puede facilitar enormemente la comprensión de algunos atributos. Uno de los elementos que permite modificar el estilo es el tamaño y color de los nodos, algo que puede resultar útil para variarlos en función del valor de alguno de sus atributos. Esto se consigue mediante el uso de las funciones de mapeo disponibles en la librería, veamos un ejemplo:

```
selector: 'node',
style: {
        'height': "mapData(weight, 10, 100, 4, 16)",
        'width': "mapData(weight, 10, 100, 4, 16)"
}
```

En este caso, se está definiendo el estilo para los nodos, como podemos observar por el elemento “selector”. En concreto, se está definiendo que el alto y ancho de los nodos varíen entre [4, 16] en función al valor de su atributo “weight”. La función mapData utilizada asignará el tamaño 4 a los nodos que tengan un valor de 10 o inferior, un tamaño 16 a todos los nodos que tengan un valor de “weight” igual o superior a 100 y los nodos que se encuentren entre medias tendrán un tamaño proporcional a su valor dentro de esta escala.

Al igual que se utilizó para modificar el tamaño de los nodos, puede emplearse para variar su color u opacidad, o también para modificar atributos propios de las aristas como el grosor. Otro elemento destacado es el “label” que permite asignar a los nodos y a las aristas una etiqueta que puede ser, por ejemplo, el valor de un atributo del mismo. De cara al uso en este proyecto, se tiene la posibilidad de variar ciertas características visuales de los nodos y las aristas en base a ciertos atributos como puede ser el valor de alguna de las métricas en el caso de los nodos o el peso en el caso de las aristas. Además, la posibilidad de añadir etiquetas permite que se pueda mostrar el nombre de usuario o el título de la página como asociado a un nodo o a una arista que permita identificarlos con facilidad.

En lo relativo a los elementos del grafo, Cytoscape propone un formato de JSON en el que existen únicamente dos grupos: nodos y aristas. En el caso de los nodos, el único atributo obligatorio es un "id" único, además, existe algunos atributos reservados que permiten determinar ciertas características como, por ejemplo, la posición del nodo (en caso de que quiera asignarse manualmente) o si puede ser "arrastrado" por el usuario de su posición original. En cuanto a las aristas, además de un "id", será necesario especificar un origen y un destino mediante el identificador de los nodos que une. Además, en ambos grupos se permite añadir libremente cualquier atributo que se desee siempre y cuando no colisionen con los atributos reservados, pudiendo añadir en este caso las etiquetas de los nodos o el valor de alguna métrica o peso.

A la vista está que el formato propuesto no es excesivamente complejo. Sin embargo, con el fin de facilitar el desarrollo se decidió investigar la existencia de algún plugin o librería que permitiera la conversión desde un objeto de NetworkX o desde alguno de los formatos soportados por el script para exportar el grafo. Por desgracia, la búsqueda no fue fructífera, optándose entonces por generar dentro del propio script un objeto JSON que cumpla con las características requeridas. Esto se logró fácilmente con la librería JSON de Python que facilita enormemente el trabajo con este tipo de objetos, además, como se vio en el apartado 4.1, la librería NetworkX ofrece un fácil acceso a todos los datos de los nodos y las aristas.

El objeto layout será el que determine como se posicionarán los nodos dentro del espacio de visualización. Como se comentó en el apartado 3.4, el layout es un elemento de sobra estudiado y utilizado en la visualización de los grafos. Probablemente, una de las mayores limitaciones que tiene la librería Cytoscape sea la poca oferta de layouts de la que dispone, más aún si se compara con la ofrecida por aplicaciones de tratamiento de grafos como Gephi o la propia Cytoscape.

En el caso de este proyecto, como ya comentamos en un capítulo anterior, los layouts más propicios son los llamados "Force-directed", ya que permiten ubicar los nodos en base a la fuerza y cantidad de sus enlaces, evitando en la medida de los posible los cruces de aristas y los solapamientos de nodos. Buenos ejemplos de este tipo de representaciones son los algoritmos propuestos por Fruchterman & Reingold [18] o por Yifan Hu [19]. Sin embargo, pese a ser algoritmos ampliamente utilizados, no se encuentran disponibles para la librería Cytoscape, siendo

el layout Cose (Compound Spring Embedder) el único de este tipo incluido por defecto en la librería y, aunque su rendimiento es bastante bueno, los resultados no son todo lo precisos que se requerían.

No obstante, entre las virtudes que proporciona el código abierto en general y Cytoscape en particular, es la posibilidad de que cualquiera puede desarrollar extensiones o plugins para la librería. Gracias a esto, se dispone de unos cuantos layouts extra a los incluidos por defectos, de entre los que se consiguió encontrar tres que cumplen con los requisitos.

- **Cose-Bilkent**: Es una variante del layout Cose, desarrollado por la universidad de Bilkent, que proporciona un rendimiento ligeramente inferior, pero con unos resultados mucho más precisos. Además, proporciona distintos elementos de configuración y personalización entre los que se encuentran el tiempo de refresco, la fuerza de las distintas gravedades, las animaciones de los nodos o el número de iteraciones a emplear para el cálculo de las posiciones.

- **Cola**: basado en algoritmos de "Force-directed", con un rendimiento superior a Cose-Bilkent, especialmente con grafos de gran tamaño y con unos resultados bastante precisos. Permite también ciertos parámetros de configuración entre los que se encuentra el espaciado entre nodos, mostrar las etiquetas de los nodos o limitar el tiempo máximo de ejecución.

- **Spread**: basado en un funcionamiento en dos fases. En la primera calcula la posición de los nodos mediante un algoritmo "force-directed", por defecto usa Cose, pero se le puede especificar otro. En una segunda fase, emplea la librería Javascript-Voronoi basada en el algoritmo de Steven J. Fortune, para expandir los nodos dentro de todo el campo de visualización disponible. De esta forma el grafo ocupa todo el campo y los espacios entre nodos se amplían permitiendo una visualización con el menor solapamiento posible. El mayor inconveniente de este layout es el rendimiento, puesto que el cálculo en doble fase aumente notablemente el tiempo y, por tanto, baje el rendimiento.

Pruebas de visualización

Una vez que estaban cubiertas las necesidades mínimas para poder visualizar los grafos con Cytoscape se propuso realizar pequeñas pruebas para poder dar por concluido el desarrollo de la segunda fase. Para ello se creó una página estática en HTML lo más sencilla posible que contuviera únicamente un contenedor donde visualizar el grafo y el código Javascript necesario para mostrarlo. Además, se añadió al script la posibilidad de volcar el grafo en el formato JSON requerido por Cytoscape a un archivo de texto plano.

El objetivo era lograr leer los datos del archivo mediante Javascript y pasárselos a la librería Cytoscape, junto con un estilo que mostrara en las etiquetas de los nodos el nombre de los usuarios, que variara el tamaño de los nodos en base a una métrica que se encontrara en los atributos y que modificara el grosor de las aristas en base a su peso. También se harían pruebas con los diferentes layouts propuestos y con los datos de wikis de distintos tamaños.

El resultado de las pruebas fue positivo, puesto que, siguiendo el proceso pautado, se consiguió la visualización del grafo dentro de la propia página HTML. Además, se extrajeron las siguientes notas sobre el rendimiento de los layouts:

- **Cose-Bilkent**: Mostró un buen rendimiento con grafos de tamaños pequeños y medios, reduciendo notablemente su rendimiento con los grafos más grandes, pero logrando igualmente mostrar el grafo. En cuanto a la calidad de la representación, los resultados eran precisos y reflejaban correctamente la fuerza de los enlaces entre los nodos. No obstante, en casos de grafos de tamaño medio-grande mostraba problemas de solapamiento.
- **Cola**: De los tres fue el que mejor rendimiento mostró, tanto con grafos pequeños, medianos y grandes, notándose solo una bajada de rendimiento con grafos de un tamaño muy grande. En lo referente a la calidad de la representación obtuvo unos resultados muy similares al Cose-Bilkent, solucionando los problemas de solapamiento que mostraba este.
- **Spread**: Mostró un rendimiento bueno con grafos pequeños, sin embargo, con grafos medianos se observó una bajada notable y con grafos grandes no llegaba siquiera a mostrarlos. En cuanto a la representación, no es comparable a los otros

dos, puesto que genera una disposición bastante diferente que permite visualizar fácilmente los enlaces entre nodos, pero no la agrupación de los mismos en base a la fuerza de sus enlaces.

Con estas pruebas se dio por concluida la segunda fase del desarrollo que consistía en la visualización del grafo generado por el script resultante de la primera fase.

4.3 Aplicación web

Recapitulando lo visto hasta ahora, se llega a esta última fase con dos módulos diferenciados. Por un lado, se tiene un script, resultado de la primera fase, que a partir del dump de una wiki puede obtener dos grafos (usuarios y páginas), calcular métricas sobre el grafo y exportarlo a distintos formatos. Como resultado de la segunda fase se dispone de una página HTML que, haciendo uso de la librería Cytoscape.js, permite leer un grafo de un archivo representado en un formato JSON propio y cambiar ciertas características de la visualización como puede ser el tamaño de los nodos, el ancho de las aristas o la disposición de la red.

Cómo nexo entre las dos partes se cuenta con la capacidad del script de exportar el grafo en el formato JSON que es capaz de entender la librería Cytoscape.js. Es decir, la herramienta ya se podría utilizar, de una forma muy rudimentaria, pero que cumpliría con su cometido. Para ello, bastaría con ejecutar el script sobre el dump a analizar, generar el grafo que se desee con las métricas que se estimen necesarias y exportarlo al formato JSON. Después se copiaría el archivo exportado (si fuera necesario) al directorio donde esté la página HTML y bastaría con abrirla para visualizar el grafo.

Sin embargo, esta forma de uso no es cómoda y va en contra de uno de los principios sobre los que se desarrolla la herramienta: su facilidad de uso. Cómo solución a esta problemática se propone implementar una solución basada en el empleo de tecnologías web, es decir, unificar los dos módulos bajo una aplicación web. El diseño del sistema constará, como es habitual, de dos partes: un lado cliente y un lado servidor. Este se puede ver en la Figura 4.1 y a continuación se desgranarán las funciones de cada parte.

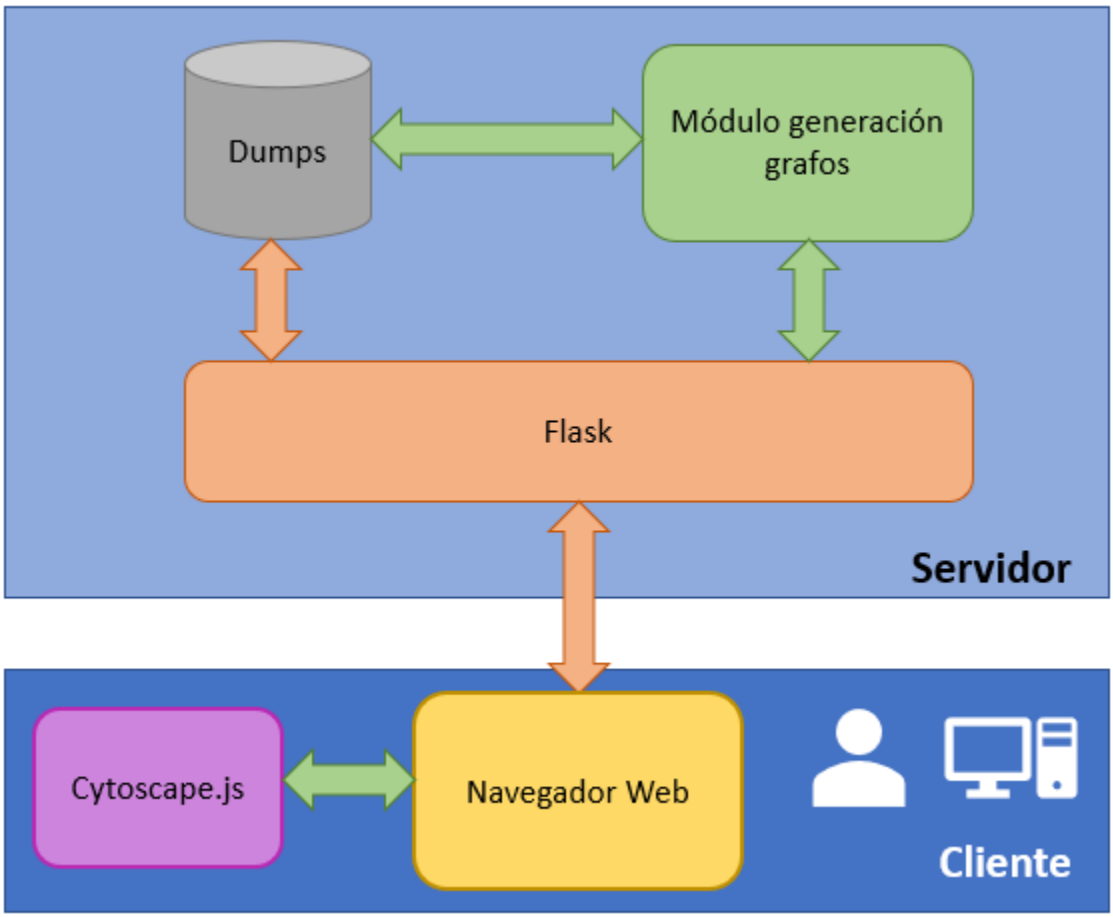

Figura 4.1 Arquitectura cliente-servidor

4.3.1 Cliente

El lado cliente será un navegador web para comunicarse con el servidor que, en primer lugar, realizará una petición de la página de inicio. Esta será una página HTML con código Javascript que se encargará de recoger las opciones elegidas por el usuario sobre las características del grafo a calcular y de la wiki sobre la que trabajar, y enviárselas al servidor para el cálculo.

Además, tendrá que ser capaz de recibir el grafo resultante del servidor y visualizarlo mediante la librería Cytoscape.js. También será capaz de gestionar las interacciones del usuario con la visualización tales como mover nodos o hacer *zoom*. Para comunicarse con el servidor se utilizará AJAX, una técnica de desarrollo web basado en el uso de Javascript y el envío de datos

en XML o JSON (como es el caso de este trabajo) de forma asíncrona, aumentando la usabilidad de la página y su rendimiento.

Entrando ya en detalles de implementación, se utilizó como base el HTML resultante de la fase anterior, puesto que ya tiene el código necesario para visualizar el grafo, y solo requiere ciertas modificaciones. Siguiendo una cierta secuencialidad de uso, será necesario añadir al diseño de la página un formulario donde el usuario pueda elegir la wiki sobre el que trabajar y las opciones deseadas para la generación del grafo.

Teniendo en cuenta las posibilidades de nuestro script se decidieron añadir los siguientes elementos al formulario:

- **Selector de wikis**: se mostrarán todas las wikis disponibles en el servidor y se permitirá elegir uno.
- **Selector de tipo de grafo**: se dejará elegir entre grafo usuario y páginas, siendo el primero aquel en el que los nodos son los usuarios y los enlaces las páginas y, la otra opción, su opuesto.
- **Slider para peso mínimo de arista**: se podrá elegir cual será el peso mínimo de las aristas.
- **Checkbox eliminar usuarios anónimos:** se podrá elegir si se desea eliminar los usuarios anónimos de la generación del grafo.
- **Checkbox ocultar etiquetas de anónimos**: en el caso de que se decida no eliminar los usuarios anónimos se podrá elegir esta opción para que no se muestre su etiqueta con el fin de reducir el ruido visual.
- **Selector de métrica**: se mostrarán todas las métricas que es capaz de calcular nuestro script. En base a la métrica elegida se variará el tamaño de los nodos, permitiendo visualizar fácilmente las diferencias de valores.
- **Selector de layout**: se permitirá elegir entre los tres layouts disponibles para decidir cuál utilizar en la visualización del grafo.
- **Botón de generación**: accionará el envío de las opciones al servidor, la generación del grafo para su posterior recepción y visualización.

Con esto, el código Javascript extraerá los valores seleccionados en cada elemento del formulario para enviarlos al servidor y esperar la respuesta. Cómo la parte de representación ya estaba hecha de la anterior fase, se puede considerar completado el lado cliente.

4.3.2 Servidor

En lo relativo a la parte del servidor web, se requiere que sea capaz de estar escuchando las peticiones que realice el usuario a través del navegador y de servir, en primera instancia, la página HTML creada con anterioridad. Además, será capaz de recibir del lado cliente las opciones elegidas por el usuario, calcular el grafo en base a las mismas, exportarlo al formato JSON y enviárselo al lado cliente para su visualización. También es recomendable, dada la naturaleza del servidor y el papel que va a desarrollar, que sea ligero, compatible en la medida de lo posible con Python y fácil de desarrollar.

Estos requisitos fueron los que propiciaron la decisión de utilizar Flask, un microframework de código abierto que facilita la creación de un servidor web en Python con una cantidad mínima de código. Cómo se puede observar, solo con su definición parece cumplir con todos los requisitos, además, el principio de desarrollo es similar al de JQuery, hacer más con menos. Flask permite trabajar con cualquier módulo de Python con total normalidad, permitiendo en nuestro caso incluir el script desarrollado en la primera fase como un módulo y poder utilizar sus funciones para generar el grafo en el propio servidor, siendo este el módulo de generación de grafos.

Para el funcionamiento de nuestro servidor será suficiente con estar escuchando en dos rutas, una la ruta raíz donde servirá la página HTML de inicio y otra donde reciba las peticiones con las opciones del usuario. En esta última deberá extraer los valores de la petición, utilizar las funciones necesarias de nuestro módulo y devolver el grafo generado en formato JSON. Además, como funcionalidad añadida, devolverá dentro del JSON el número de nodos y aristas, para que se le muestre al usuario como información complementaria a la visualización.

Sin embargo, aunque la solución esté completada, aún quedan algunas cuestiones que solventar. En concreto queda saber cómo se obtendrán los dumps disponibles para mostrárselo al usuario. Una solución puede ser añadir manualmente a la página HTML cada vez que haya un dump nuevo. No obstante, esta solución es poco flexible, más costosa y resta autonomía al servidor y, por ende, a la propia herramienta. Como solución a esta problemática Flask incluye, dentro de su framework, un lenguaje de plantillas para Python: JINJA.

Esta herramienta nos permite crear plantillas de páginas HTML con partes que se rellenan en tiempo de ejecución. Gracias a esta funcionalidad podemos rellenar el selector de wikis con los dumps que haya disponibles dentro del servidor en el momento que nos soliciten la página. Dicho de otra forma, si añadimos el dump de una nueva wiki al servidor todas las peticiones que lleguen después tendrán la nueva opción disponible en el selector, dotando a la herramienta de una flexibilidad y transparencia total para añadir nuevas wikis. Con esta solución podemos dar por completado el servidor web y, por consecuencia, la tercera y última fase del desarrollo de la herramienta.

Capítulo 5 - Caso de estudio

Una vez terminada la fase de desarrollo, se tiene como resultado la herramienta final del proyecto. Recapitulando lo ya visto, se tiene una aplicación web cuya lógica se separa por un lado en el servidor web encargado de comunicarse con el cliente, calcular el grafo con las opciones solicitadas y enviar el resultado como respuesta. En el lado cliente se tiene una página web con código Javascript que se encargará de recoger las opciones elegidas por el usuario, enviarlas al servidor, recibir la respuesta y visualizarla.

La parte dedicada al funcionamiento interno de la aplicación es de sobra conocido, por tanto, veamos cómo es el uso de la herramienta desde el punto de vista del usuario. Lo primero que nos encontramos es la página inicial, cómo se puede ver en la Figura 5.1. Compuesta por una barra de herramientas con distintas opciones y un área en blanco que será donde se visualice el grafo.

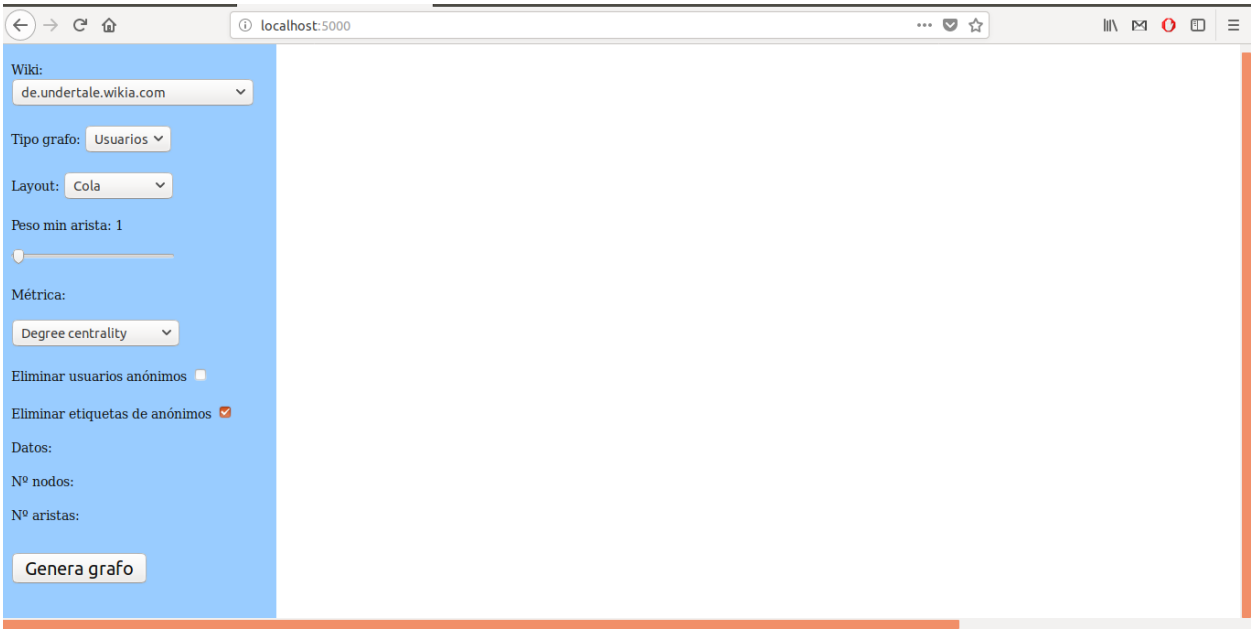

Figura 5.1 Página inicial de la herramienta

Ahora veamos un caso de uso sobre una de las wikis disponible en el servidor. Si generamos el grafo con las opciones por defecto nos dará un resultado como el de Figura 5.2. Se

puede apreciar un grafo de grandes dimensiones y difícil estudio. Para solucionarlo vamos a aplicar uno de los filtros que permite la herramienta: eliminar los usuarios anónimos. Esto es de gran utilidad ya que el comportamiento de este tipo de usuarios es un tanto indefinido ya que la forma en que tiene Wikia de registrarlos es por su dirección IP. Esto puede significar que una misma persona venga representada por múltiples usuarios anónimos, en caso de que se conectara desde distintas IPs, o, al contrario, que varias personas vengan representadas por un único usuario anónimo.

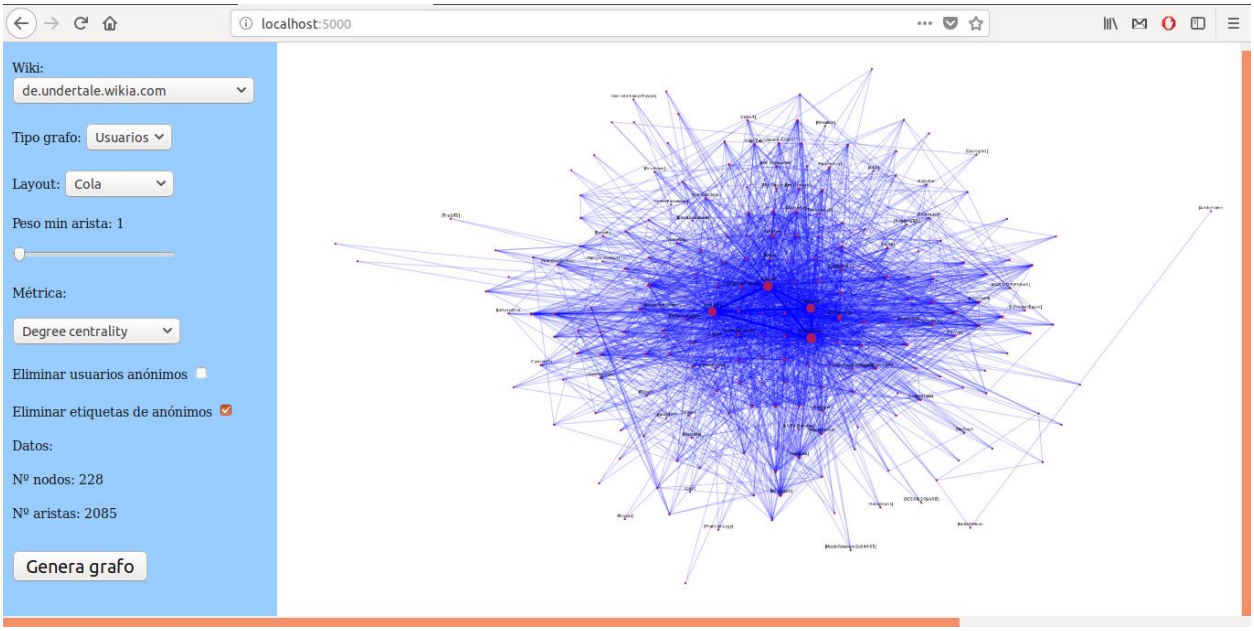

Figura 5.2 Grafo con opciones por defecto

Una vez realizado el filtrado de usuarios anónimos, podemos ver el resultado en la Figura 5.3, observamos que las dimensiones del grafo han disminuido notablemente, reduciéndose el número de usuarios a algo menos de una tercera parte y el de aristas a una quinta parte. Este cambio nos genera un grafo mucho más legible, pero al mismo tiempo plantea una duda: si el número de usuarios registrados es tan pequeño en proporción al total, ¿se pueden obtener conclusiones sobre el comportamiento real de la comunidad que conforma la wiki? Esta cuestión es difícil de responder dado que, como ya explicamos, no podemos determinar a cuantas personas reales representan los usuarios anónimos. Por tanto, seguiremos analizando sobre los usuarios registrados.

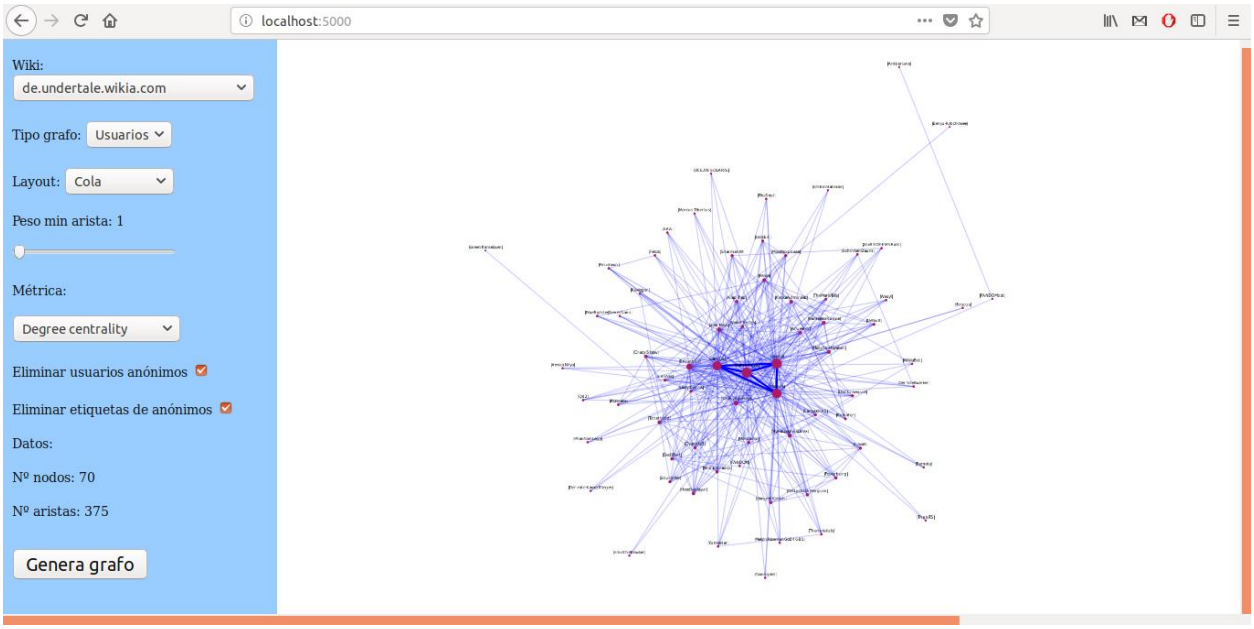

Figura 5.3 Grafo sin usuarios anónimos

Veamos ahora que sucede si consideramos que todas las aristas de peso inferior a tres no son relevantes. Dicho de otro modo, si dos usuarios no han editado al menos tres páginas en común podemos considerar que no tienen una relación suficientemente fuerte como para tenerse en cuenta en el estudio. Para ello aplicamos el filtrado de aristas y establecemos el peso mínimo en tres. En la Figura 5.4 observamos el resultado del filtrado, los usuarios de este grafo podrían ser considerados como los usuarios más activos de la wiki y los que más han interactuado entre sí.

Una vez limpiado el grafo pasemos a analizarlo. Observamos que de entre los usuarios resultantes hay cuatro especialmente importantes, esta deducción se obtiene por varias vías. Lo primero, tienen un tamaño de nodo superior al resto lo que implica que el grado de centralidad es alto en ellos. Además, podemos ver que les unen aristas de un grosor superior al resto, indicando que el peso de las mismas es elevado. Por último, gracias al layout guiado por fuerzas, los cuatro usuarios se sitúan en el centro del grafo, bastante cerca los unos de los otros, mientras que el resto de los usuarios se ubican en la periferia mostrando unos enlaces más débiles con el resto del grafo.

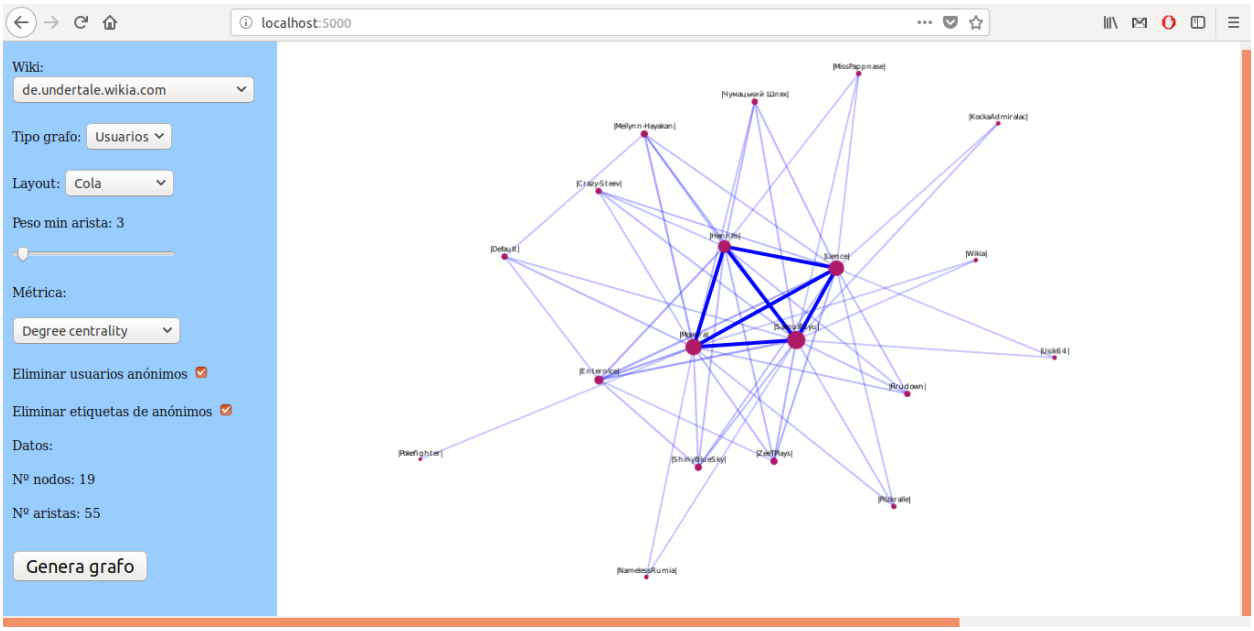

Figura 5.4 Grafo con aristas filtradas por peso

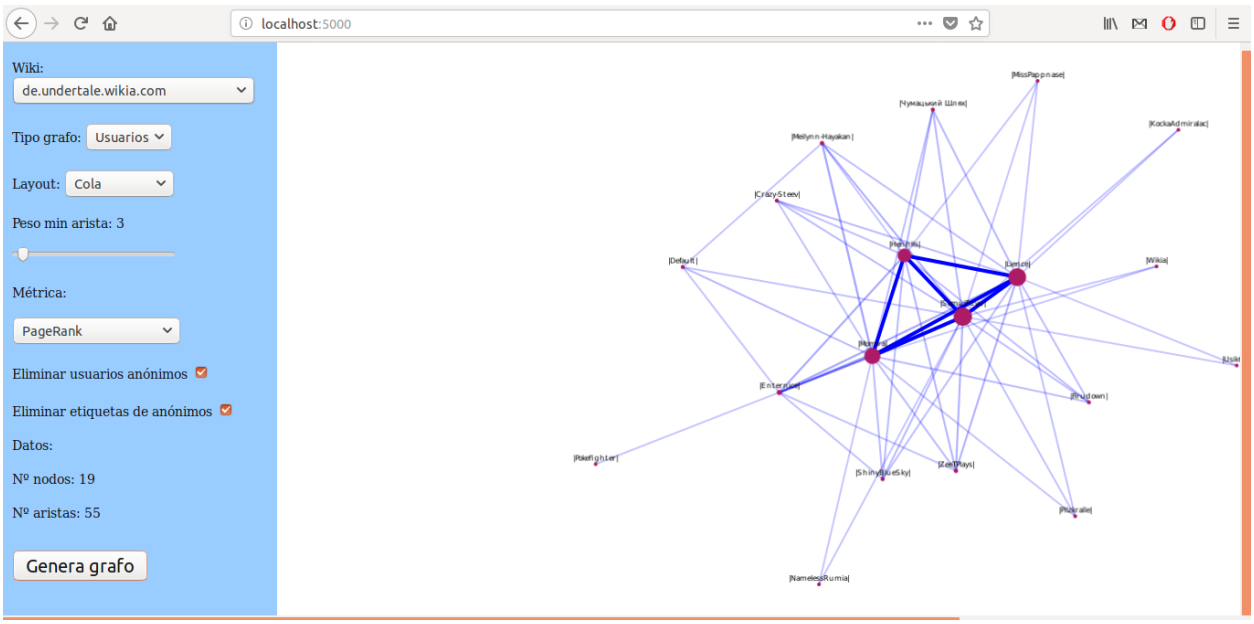

Figura 5.5 Grafo con Page Rank

Por último, vamos a modificar la métrica con la que medir la centralidad de los nodos. En este caso elegiremos PageRank para poder observar como de centrales son los usuarios a nivel global. En el resultado, que se observa en la Figura 5.5, vemos un resultado bastante similar al anterior en el que se ha reducido más el tamaño de los nodos periféricos, mientras que los cuatro

usuarios centrales mantienen un tamaño similar. Esto nos demuestra que esos cuatro usuarios son lo que se podría considerar el "núcleo duro" de la wiki.

Con este caso de uso se ha podido visibilizar como, a partir del dump de una wiki, se puede localizar a los usuarios más importantes de la misma. Además, se puede ver cómo el uso de la herramienta resulta sencillo e intuitivo. También, cumple sobradamente los requisitos establecidos a nivel de funcionalidad, dando como resultado una herramienta con un alto potencial de uso. Por último, es destacable el carácter web de la aplicación, ya que permite su uso sin necesidad de instalación y de forma independiente al sistema operativo confiriéndole una versatilidad que es siempre bien recibida en una herramienta de estas características.

5.1 Limitaciones

Durante las fases de desarrollo y pruebas de la aplicación se encontraron algunas limitaciones en el uso que no fue posible corregir. Esta falta de soluciones se debe principalmente a que todas las limitaciones conocidas son fruto del tamaño de la wiki a analizar, requiriendo mayor número de recursos y tiempo de ejecución, haciendo que el funcionamiento de la wiki se vea mermado y la experiencia de usuario reducida en ciertos casos. Aunque a lo largo de la memoria ya han sido tratadas, se agrupan bajo el siguiente listado todas las limitaciones conocidas:

- Grafo asociado a páginas: este grafo es, normalmente, de un tamaño muy superior al grafo asociado a los usuarios. Se ha observado que esta diferencia genera que incluso en wikis de un tamaño pequeño o mediano, haya un gran contraste de rendimiento entre el grafo asociado a páginas y el asociado a usuarios.
- Visualización de redes de tamaño mediano: se ha observado que el layout Spread tiene problemas de rendimiento a la hora de mostrar wikis de este tamaño.
- Visualización de redes de tamaño grande y muy grande: se ha observado que el layout Cose-Bilkent muestra problemas de rendimiento con wikis de tamaño grande y no pudiendo mostrar aquellas de tamaño muy grande. Además, se han detectado problemas de rendimiento del layout Cola con wikis de un tamaño muy grande. Por su parte, el layout Spread no es capaz de visualizar wikis de tamaño grande o muy grande.

Capítulo 6 - Conclusiones y trabajo futuro

Para finalizar podemos concluir que la producción colaborativa de conocimiento es, en teoría, una forma de producción descentralizada, autónoma y altruista que se presenta como un modelo alternativo al hegemónico modelo basado en la competitividad. Sin embargo, como hemos visto, este tipo de producción genera ciertas dudas a la hora de llevarla a la práctica. Por ejemplo, una de las problemáticas que se le presuponen es la desigual carga de trabajo dentro de la propia comunidad de producción. Para poder detectar problemas de este tipo y, en caso de confirmarse, buscar una solución deben de realizarse estudios sobre comunidades ya existentes. Es aquí donde entra en juego la herramienta desarrollada en este proyecto.

Tal y cómo se explicó esta herramienta pretende facilitar el estudio de uno de los tipos de comunidad de producción colaborativa más extendida: las wikis. Con esta premisa en mente se ha desarrollado una herramienta prototipada para funcionar con Wikia debido a que su condición de host de wikis nos permite el estudio de una amplia variedad de wikis de diversos tamaños. Además, la aplicación se apoya en el análisis de redes para poder obtener métricas de gran valor para el estudio de la comunidad, así como para diseñar la mejor forma de visualización y representación de esta, facilitando enormemente el análisis.

Como resultado se obtuvo una aplicación funcional que cumple con su cometido. Además, debido a ciertas elecciones técnicas se dota a la herramienta de valores añadidos como ser fácilmente extensible, integrable y usable. Las dos primeras características se logran, en parte, por estar desarrollada sobre Python y Javascript, dos lenguajes extremadamente populares y que cuentan con una amplia variedad de librerías que permiten aumentar su funcionalidad. En lo respectivo a la usabilidad de la herramienta, esta se debe a la decisión de diseñarla cómo aplicación web y bajo una interfaz intuitiva, haciendo su uso accesible a la mayoría de los usuarios.

Como conclusión, podemos determinar que la herramienta, si bien es y será objeto de mejora, cumple de forma notable con los objetivos planteados inicialmente. Siendo, por tanto, un pequeño, pero funcional aporte al estudio de la producción colaborativa.

6.1 Trabajo futuro

El presente proyecto por su carácter como Trabajo Fin de Máster, tiene unos recursos y un tiempo limitado. Cómo consecuencia, a veces se tiene que renunciar a implementar ciertas mejoras que surgen durante la planificación y el desarrollo de la herramienta. No obstante, con el fin de que futuros proyectos puedan mejorar y aumentar la funcionalidad de la aplicación resultante se listará una serie de propuestas de ampliación:

- Incluir dentro de la aplicación las herramientas de descarga y limpieza de dumps, permitiendo al usuario analizar una wiki a su elección proporcionando únicamente la url de la misma.
- Mostrar varias métricas de un nodo al mismo tiempo. Se podría implementar un comportamiento en el que, al seleccionar un nodo en la visualización, muestre los valores de todas las métricas asociadas al mismo.
- Realizar una precargar de los grafos de las wikis disponibles. Bastaría con calcular los grafos de todos los dumps disponibles en el servidor y almacenarlos a modo de caché, evitando tener que generar el grafo a cada petición.
- Aumentar el número de métricas disponibles. Para ello se podría partir por implementar algunas de las métricas descartadas por falta de tiempo y por costes de implementación.
- Permitir al usuario elegir características de visualización tales como el color de los nodos y aristas o el tamaño máximo y mínimo de los nodos.

Chapter 6 - Conclusions and future work

To finalize, we can conclude that collaborative production of knowledge is, in theory, a decentralized, autonomous and altruistic form of production that is presented as an alternative model to the hegemonic model based on competitiveness. However, as we have seen, this type of production generates certain doubts when it comes to putting it into practice. For example, one of the problems that are presupposed is the unequal workload within the production community itself. To detect problems of this type and, if confirmed, seek a solution should be conducted studies on existing communities. This is where the tool developed in this project plays its role.

As explained, this tool aims to facilitate the study of one of the most widespread types of collaborative production community: wikis. With this premise on mind, a prototyped tool has been developed to work with Wikia because its wikis host status allows us to study a wide variety of wikis of different sizes. In addition, the application relies on network analysis to obtain high-value metrics for the study of the community, as well as to design the best way to visualize and represent it, helping the analysis.

As a result, a functional application that fulfills its purpose was obtained. In addition, due to certain technical choices, added values such as being easily extensible, integrable and usable have been given to the tool. The first two features have been achieved, in part, by being developed on Python and Javascript, two extremely popular languages with a wide variety of libraries that increases their functionality. Regarding to usability of the tool, it is achieved due to the decision of designing the tool as a web application and under an intuitive interface, making its use accessible to most users.

In closing, we can determine that the tool, although it is and will be the object of improvement, fulfills in a remarkable way the objectives initially proposed. Being, therefore, a small, but functional contribution to the study of collaborative production.

6.1 Future work

The present project has limited resources and time given its condition of Master's Final Project. Consequently, one must sometimes give up implementing certain improvements that are originated during the planning and development of the tool. However, we suggest the next list of proposals to improve and increase the functionality of the resulting application regarding to future projects:

- To include within the application the tools for downloading and cleaning dumps, allowing the user to analyze a wiki of their choice by providing only the URL of the same.
- To show several metrics of a node at the same time. A behavior could be implemented in which, when selecting a node in the visualization, it shows the values of all the metrics associated to it.
- To make a preload of the graphs of the available wikis. It would be enough to calculate the graphs of all the available dumps in the server and store them as a cache, avoiding having to generate the graph for each request.
- To increase the number of metrics available. To do so, we could start by implementing some of the metrics discarded due to lack of time and implementation costs.
- To allow the user to choose display characteristics such as the color of the nodes and edges or the maximum and minimum size of the nodes.

Bibliografía

[1] BENKLER, Yochai; NISSENBAUM, Helen. Commons-based peer production and virtue. Journal of political philosophy, 2006, vol. 14, no 4, p. 394-419.

[2] WU, Fang; WILKINSON, Dennis M.; HUBERMAN, Bernardo A. Feedback loops of attention in peer production. En Computational Science and Engineering, 2009. CSE'09. International Conference on. IEEE, 2009. p. 409-415.

[3] NIELSEN, Jakob. The 90-9-1 rule for participation inequality in social media and online communities. Online: https://www. nngroup. com/articles/participation-inequality, 2006.

[4] LEUF, Bo; CUNNINGHAM, Ward. The Wiki way: quick collaboration on the Web. 2001.

[5] JIMENEZ-DIAZ, Guillermo; SERRANO, Abel; ARROYO, Javier. A Wikia census: motives, tools and insights. In Proceedings of the 14th International Symposium on Open Collaboration (OpenSym '18). ACM Press, Article 2, 6 pages.

[6] ZHOU, Tao, et al. Bipartite network projection and personal recommendation. Physical Review E, 2007, vol. 76, no 4, p. 46-115.

[7] BOCCALETTI, Stefano, et al. Complex networks: Structure and dynamics. Physics reports, 2006, vol. 424, no 4-5, p. 175-308.

[8] BARABÁSI, Albert-László; PÓSFAI, Márton. Network science. Cambridge university press, 2016.

[9] BORGATTI, Stephen P.; EVERETT, Martin G. A graph-theoretic perspective on centrality. Social networks, 2006, vol. 28, no 4, p. 466-484.

[10] FREEMAN, Linton C. Centrality in social networks conceptual clarification. Social networks, 1978, vol. 1, no 3, p. 215-239.

[11] SUN, Jimeng; TANG, Jie. A survey of models and algorithms for social influence analysis. En Social network data analytics. Springer, Boston, MA, 2011. p. 177-214.

[12] BONACICH, Phillip. Some unique properties of eigenvector centrality. Social networks, 2007, vol. 29, no 4, p. 555-564.

[13] BONACICH, Phillip. Power and centrality: A family of measures. American journal of sociology, 1987, vol. 92, no 5, p. 1170-1182.

[14] PAGE, Lawrence, et al. The PageRank citation ranking: Bringing order to the web. Stanford InfoLab, 1999.

[15] KLEINBERG, Jon M. Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM), 1999, vol. 46, no 5, p. 604-632.

[16] MISLOVE, Alan, et al. Measurement and analysis of online social networks. En Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. ACM, 2007. p. 29-42.

[17] HERMAN, Ivan; MELANÇON, Guy; MARSHALL, M. Scott. Graph visualization and navigation in information visualization: A survey. IEEE Transactions on visualization and computer graphics, 2000, vol. 6, no 1, p. 24-43.

[18] FRUCHTERMAN, Thomas MJ; REINGOLD, Edward M. Graph drawing by force-directed placement. Software: Practice and experience, 1991, vol. 21, no 11, p. 1129-1164.

[19] HU, Yifan. Efficient, high-quality force-directed graph drawing. Mathematica Journal, 2005, vol. 10, no 1, p. 37-71.